Application of Particle Swarm Optimization Based on Neural Network for Artillery Range Prediction

Y.-W. Chen*, Y.-L. Lee, C.-C. Kung*****

**School of Defense Science,
Chung Cheng Institute of Technology, National Defense University,Taiwan
(e-mail: chenyiwei530@yahoo.com.tw).
**Department of Power Vehicle and Systems Engineering,
Chung Cheng Institute of Technology, National Defense University,Taiwan
(e-mail: dragon0913@yhoo.com.tw)
***Department of Mechatronic, Energy and Aerospace Engineering,
Chung Cheng Institute of Technology, National Defense University,Taiwan
(e-mail: cckung@ndu.edu.tw)*

Abstract: The firepower of artillery is one of main factors to influence the war effectiveness. Traditionally, the army utilizes the firing table to modify the artillery range, but the fabrication of firing table of artillery costs a lot of time and ammunition. In this study, some firing data of artillery are utilized to train the back-propagation neural network for artillery range prediction. Particle swarm optimization is utilized to increase the training speed of neural network and avoid getting stuck in local extreme. Besides, the orthogonal array is used to decrease the requirement of firing data and the proposed method is compared with the traditional back-propagation neural networks. Simulation results verify that the proposed method can not only increase the training speed of neural network but also have the satisfied performance of range prediction, and the mean absolute percentage error can approach to 1.173%. The proposed method in this paper is usable for artillery range prediction and feasible for application in the army.

Keywords: Neural network, particle swarm optimization, artillery, orthogonal array.

1. INTRODUCTION

The artillery firepower plays an important role in the war. The military research for each nation is always trying to improve the firing accuracy of artillery and the artillery fabrication technique has a lot of development in recent years. Generally, the firing table of man-made method is used to obtain the fire-control data of artillery to modify the range, but the fabrication process costs huge amount of time and money. For the development of ballistics, aerodynamics, mathematics and computer technique, the fabrication process of modern firing table can decrease the amount of ammunition and time, but the method depends on the accurate mathematical ballistic equation and aerodynamic property. Besides, it is the tendency to buy the overseas artillery and use the self-made ammunition, but more errors may produce in firing. Some literatures utilized the regression analysis method or neural network to build the artillery equations for the prediction of impact point, ricochet, artillery range, and modeling the fire tables. (Fei et al., 2008; Sherif, 1985; Su, 2007).

Neural network is usually used extensively for non-linear mapping without the complicated mathematical model. For example, it can be used on fingerprint verification (Ala et al., 2013), fault detection and isolation in industrial control valve (Ahmed et al., 2013), and the control of distillate composition of binary distillation column (Amit et al., 2013). It is proved the back-propagation neural network (BPN) can approach any nonlinear continuous function at any precision (Gross, 1988). Besides, neural network is also usually combined with design of experiments and the experimental samples are used to be the training data for building a network model to make prediction, interpolation, extrapolation and parameter optimization (Wang et al., 1998; Chang et al., 2007; Chang and Tsai, 2008; Dedy et al., 2008). The method can not only reduce the need of experimental data but obtain the satisfied result.

Particle swarm optimization (PSO) was often used to optimize the parameters of BPN to improve the local optimum and obtain the better performance on prediction and convergence (Pian et al., 2012; Chang et al., 2007). Some literature indicated PSO combined with BPN was superior to genetic algorithm combined with BPN and the traditional neural network (Zhang, 2011). Besides, the application of PSO was developed extensively. (Yongzhong et al., 2011) proposed a novel evolutionary strategy-based PSO approach which was dependent on a BP neural proportional integral derivative (PID) controller. The results showed that the proposed method could enhance the diversity of swarms, considerably improve the global convergence efficiency and outperform the PSO algorithm. PSO was also utilized to investigate the optimized operator parameterization for a hybrid texture analysis system (Peters et al., 2007).

In this paper, PSO is used to accelerate the training speed of BPN and improve the local optimum by optimizing the initial weight and bias. BPN is utilized to predict the artillery range and the application of orthogonal array can reduce the training samples of neural network.

2. APPLICATION OF PSO BASED ON BPN

2.1 Back-Propagation Neural Network (BPN)

The algorithm of neural network is to imitate the nervous system of organism. The back-propagation neural network (BPN) is composed of input layer, hidden layer and output layer. The training process contains the forward pass, error computation and error back-propagation. The configuration of BPN is shown in Fig. 1.

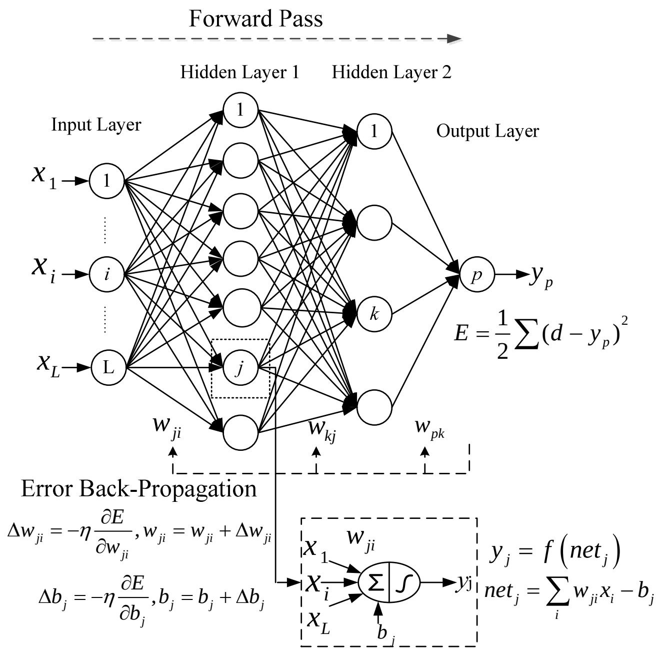

Fig. 1. The configuration of BPN.

During the forward pass, a neuron driven by the input signal would produce the output (y), and which is different from the expected output (d) will produce the error. The error function (E) is minimized by the application of the gradient steepest descent method. The error signals are then back propagated through the network from output layer to input layer called weight modification ($w_{ji} = w_{ji} + \Delta w_{ji}, b_j = b_j + \Delta b_j$). The parameters of neural network such as hidden layers, neurons, learning rate (η) and transfer function are modified properly during the training process. The parameters of neural network will influence the computing rate, prediction accuracy and convergence condition. One hidden layer can solve the common problem and two hidden layers are used to solve the complicated problem. The neuron quantity of input layer and output layer are decided by the variables of input and output. The neuron quantity of hidden layer will influence the learning efficiency of neural network. If the quantity is too small, it is difficult to converge for neural network, or the trained neural network is not strong enough and has the poor fault-tolerance (Chang et al., 2007).

2.2 Particle Swarm Optimization (PSO)

PSO is proposed firstly by Eberhart and Kennedy and the basic concept is based on the simulation of animal social behavior from birds foraging (Eberhart and Kennedy, 1995, a-b). 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. During the evolutionary process, each particle searches continuously for the optimal solution in problem space and memorizes the path. Besides, the optimal solution between particles is considered. If the particle swarm is composed of m particles, and each particle searches for the optimal solution on M dimension space. The position of particle i is denoted by $X_i = (x_{i1}, x_{i2}, \cdots, x_{iM})^T$ and the velocity is denoted by $V_i = (v_{i1}, v_{i2}, \cdots, v_{iM})^T$, $1 \le i \le m$, $1 \le j \le M$. The position of optimal solution is denoted by $P_i = (p_{i1}, p_{i2}, \cdots, p_{iM})^T$, and the position of global optimal solution is denoted by $P_g = (p_{g1}, p_{g2}, \cdots, p_{gM})^T$. Each particle will update the position and velocity according to the following equations:

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_1 \left(p_{ij}(t) - x_{ij}(t)\right) + c_2 r_2 \left(p_{gj}(t) - x_{ij}(t)\right) \quad (1)$$

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

Where t denotes the iteration count, w is inertia weight, r_1, r_2 are random numbers between 0 and 1, and c_1, c_2 are learning factors which are positive constants.

2.3 PSO-BP Neural Network

BPN has a good performance in accuracy, but the weaknesses are slow convergence speed, getting stuck in local extreme and producing the instable network. PSO is utilized to search for the initial weight and bias of neural network to increase the convergence speed and improve the local optimum in this paper. The gradient steepest descent method is used to modify gradually the weight and bias to minimize the error function of BPN. Fig. 2 is the flow chart of PSO-BP neural network algorithm and the operation steps are explained as follows:

Step 1: Initialize PSO and BPN parameters. The position and velocity of M-dimension particle are random in the domain [0, 1] and the M-dimension particle is transferred to the initial weight and bias of BPN. The particle's size is equal to the quantity of initial weight and bias. The linear decreasing inertia weight equation is utilized in Eq. 3 (Kennedy and Eberhart, 1999) and w_1, w_2 are separately 0.9, 0.4. The learning factors are changed dynamically by using Eq. 4 and Eq. 5 (Asanga et al., 2004). c_{1in}, c_{2in} are separately 0.1, 0.6,

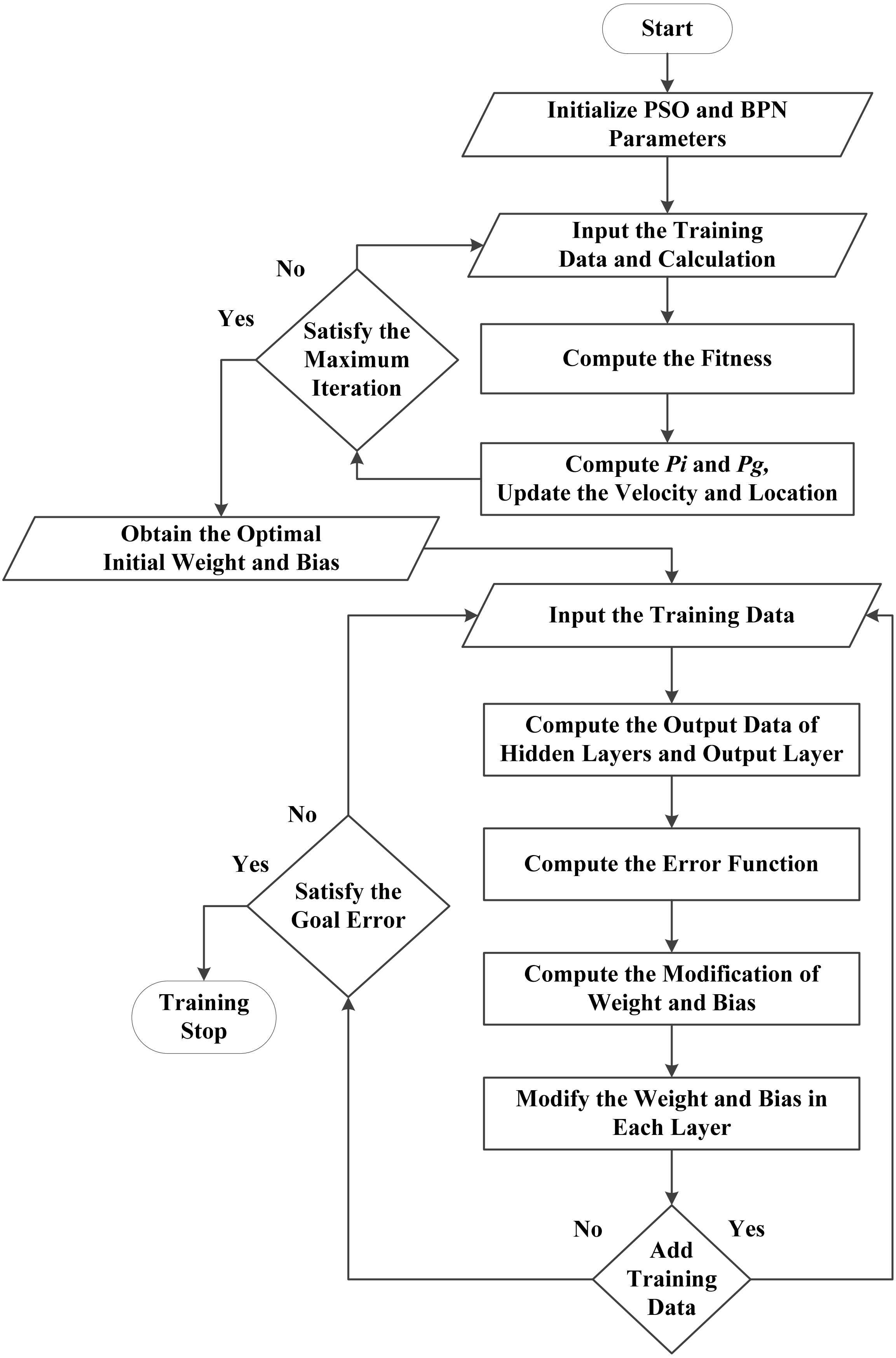

Fig. 2. Flow chart of PSO-BP neural network.

c_{1en}, c_{2en} are separately 0.5, 1. Population size: 100; Iteration: 200. Hidden layers: 2; Hidden layer neurons: (7, 4); Learning rate: 0.1; Transfer function: sigmoid function. Mean absolute error is used as the cost function and the goal error is 0.007.

$$w = \frac{(w_1 - w_2)(T_{\max} - t)}{T_{\max}} + w_2 \tag{3}$$

$$c_1 = (c_{1en} - c_{1in})\frac{t}{T_{\max}} + c_{1in} \tag{4}$$

$$c_2 = (c_{2en} - c_{2in})\frac{t}{T_{\max}} + c_{2in} \tag{5}$$

w_1 is the maximum initially and w_2 is the minimum at the end. t, $T_{\max}$ denote separately the iteration count and maximum iteration count. c_{1in}, c_{2in} are the minimal constants initially and c_{1en}, c_{2en} are the maximal constants at the end.

Step 2: Input the training data gradually and calculate the equations from Eq. 6 to Eq. 10. The operation step above is repeated until all data are computed and *n* is the data number.

$$net_j^n = \sum_i w_{ji} x_i^n - b_j \tag{6}$$

$$y_j^n = f\left(net_j^n\right) \tag{7}$$

$$net_k^n = \sum_j w_{kj} y_j^n - b_k \tag{8}$$

$$y_k^n = f\left(net_k^n\right) \tag{9}$$

$$y_p = net_p^n = \sum_j w_{pk} y_k^n - b_p \tag{10}$$

Step 3: Compute the fitness of particle and the fitness function is defined as mean absolute error as follows:

$$Fitness = \frac{\sum_{n=1}^{N} \left|\left(d^n - y_p^n\right)\right|}{N} \tag{11}$$

d^n is the expected value, y_p^n is the output value of output layer, N is the number of training data.

Step 4: Repeat step 2 to step 3 until all particles are computed (m=100). Compute the history optimal position of each particle, and if the position has the minimal fitness in the swarm, it will be the global optimal position, and then, update the velocity and position.

Step 5: Repeat steps from step 2 to step 4 until the maximum iteration is satisfied.

Step 6: Input the optimal initial weight and bias obtained by PSO to train the BP neural network.

3. ARTILLERY FIRING DATA AND ORTHOGONAL ARRAY

3.1 Artillery Firing Data

Angle of departure, muzzle velocity, air temperature, air pressure, wind velocity, wind direction and relative humidity are the major variables to influence the artillery range deeply. The artillery firing data used in this paper come from 40mm/L70 firing table of original equipment manufacturer made by huge amount of firing tests. It has two types: one is anti-craft and the other one is ground. The artillery is middle-caliber and the diameter of warhead is 40mm. The artillery firing data used here are adopted directly from the ground firing table and the data scope is as follows: angle of departure (^{0}): 1~30; muzzle velocity (m/s): 1000~1010; air temperature (^{0}C): 5~39; relative humidity (%): 50~100; air pressure (mba): 1002~1019; wind velocity (m/s): 1.7~6.3. The wind contains the following wind and the cross wind. The wind direction has two types: one is the downwind direction which has the same direction with the projectile trajectory, and the other one is the upwind direction. The standard muzzle velocity is 1005 (m/s) and the data range is defined by the fabrication deviation of ammunition. The atmosphere data range is defined by Taiwan climate condition.

3.2 Orthogonal Array

The utilization of orthogonal array in this paper is to reduce the requirement of artillery firing data and hope to obtain the satisfied result. It is the research purpose to develop a prediction model of artillery range by using less firing data. The property of orthogonal array is the same effective information as the full factorial experiment. The scheme of orthogonal array is geometrically balanced and statistically independent. Because the correlation between the artillery range (output) and the variables (input) are very complex, the three-level design are used for each variable except that the wind direction is in two-level design. The investigated parameters and levels are shown in Table 1, and $L_{36}(2^2 \times 3^7)$ orthogonal array is utilized in Table 2.

4. SIMULATION AND DISCUSSION

In this paper, 82 artillery firing data are used to build and evaluate the neural network model (Table 3). These firing data are divided into three parts. First part, the 36 artillery firing data arranged in $L_{36}(2^2 \times 3^7)$ orthogonal array are utilized to train the PSO-BP neural network (O.A. PSO-BPN) and traditional BPN (O.A. BPN). Second part, another 36 artillery firing data are dispersed and widespread in the data scope, and that are used to train the second traditional BPN. Third part, the residual 10 artillery firing data are chosen randomly in the data scope and that are used to evaluate the performance of these method. For the operation of neural network, the dimensionality of input data (variables: C～I) and output data (artillery range) are 7 and 1, respectively. On the following wind, the wind velocity would be positive/negative if the wind direction is downwind/upwind. The same condition is on the cross wind.

Table 1. The investigated parameters and levels

Variables	Levels		
	1	2	3
A: Follwing Wind Direction	Upwind	Downwind	
B: Cross Wind Direction	Upwind	Downwind	
C: Angle of Departure (^{0})	1	15	30
D: Muzzle Velocity (m/s)	1000	1005	1010
E: Relative Humidity (%)	50	75	100
F: Air Pressure (mba)	1002	1010	1019
G: Air Temperature (^{0}C)	5	22	39
H: Following Wind Velocity (m/s)	1.7	4	6.3
I: Cross Wind Velocity (m/s)	1.7	4	6.3

Table 2. $L_{36}(2^2\times3^7)$

No.	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

All of the firing data are needed to be normalized for training the neural network and the normalization equation is as follows:

$$X_2 = \frac{X_1 - X_{\min}}{X_{\max} - X_{\min}} \times \left(D_{\max} - D_{\min}\right) + D_{\min} \tag{12}$$

X_1, X_2 : the original data and normalized data separately; $X_{\min}$, $X_{\max}$: the original data of minimum and maximum separately; $D_{\min}$, $D_{\max}$: the minimum and maximum separately in [0,1] domain, and $D_{\min}$=0.1, $D_{\max}$=0.9 in this study.

Table 4 is the comparison of training time and prediction accuracy for three neural network models. Each model is repeated by five times and the mean absolute percentage error (MAPE) is the index to evaluate the prediction accuracy of neural network model. MAPE is defined as follows:

$$MAPE = \frac{1}{M}\sum_{k=1}^{M}\left|\frac{x(k) - \hat{x}(k)}{x(k)}\right| \times 100\% \tag{13}$$

$x(k)$ is the range of artillery firing data, $\hat{x}(k)$ is the predicted range of model, M is the number of samples.

From the Table 4, BPN costs the most training time and gets the worst prediction result. This condition will be improved greatly when the orthogonal array is utilized shown in O.A. BPN and O.A. PSO-BPN. The results explain two chief advantages by using the orthogonal array. One is the neural network can learn quickly. The other one is the characteristic of orthogonal array in statistics can increase the prediction accuracy of model well. In addition, the training time of neural network will be shortened obviously when the initial weight and bias is optimized by PSO. Figure 3 is the comparison of convergence process of three neural network models and each one uses the shortest training time in five repeats. Table 5 is the error comparison of prediction and truth for three neural network models. The predicted range of three neural network models is the mean of five repeats. From the analysis of MAPE in Table 5, the proposed method (O.A. PSO-BPN) has the best prediction accuracy and O.A. BPN is close to O.A. PSO-BPN. The BPN has the worst prediction accuracy and it is easy to produce the bigger errors (>4%) shown in No. 4, 8 and 10. It means the prediction model of artillery range is not robust even though the training data of BPN are dispersed and widespread in the data scope. For O.A. PSO-BPN and O.A. BPN, all of the errors are <3%, and the ratio of errors <1% are 6/10 and 4/10, respectively. The description verifies the advantages of orthogonal array and PSO. Table 5 shows that the application of PSO and O.A. improve the prediction accuracy of model undoubtedly. The MAPE can approach to 1.173%. By way of the observation in Table 4, 5 and Figure 3, it is not difficult to understand that the faults of BPN are the slow convergence speed and getting stuck easily in local extreme due to the gradient steepest descent method. The global searching ability of PSO can overcome the weaknesses. From the analysis and discussion above, the proposed method in this paper has the best performance on prediction accuracy and training speed of neural network.

5. CONCLUSIONS

The development of artillery firepower is always the research topic for the national defense industry, and that is also one of key points to influence the war result. Traditionally, the prediction of artillery range depends on the firing table, but the fabrication of firing table costs a lot of time and money. In this study, the prediction model of artillery range based on PSO-BPN and orthogonal array has the satisfied performance compared with the traditional BPN. Simulation results show the orthogonal array can improve both of the prediction accuracy and training time of neural network. The application of PSO can accelerate the training speed of neural network obviously and avoid getting stuck in local extreme. The mean absolute percentage error of PSO-BPN can approach to 1.173%. The proposed method in this paper can cost less time to build the prediction model of artillery range, and it is feasible to be applied in the army.

The fabrication of firing table is not easy, and it is the tendency to buy the overseas artillery and use the self-made ammunition. The errors in firing may also increase gradually year by year because the artillery will become elderly with time. On the basis of the reasons above, the direction of future development is trying to utilize the firing data produced by the army exercise or firing tests in each year for building a popular-updating prediction model of artillery range. One of the main factors to determine the prediction accuracy of model in this paper is the convergence precision of BPN. It is proven that BPN can converge at any precision but the over-fitting should be avoided. The application of PSO overcomes the faults of BPN as the slow convergence speed and getting stuck in local extreme. Therefore, it is finite to improve PSO-BPN by using other algorithm of artificial intelligence to increase the prediction accuracy of model. The quantity of training samples is the other main factor to influence the performance of neural network. It is feasible to add the training samples properly based on the design of orthogonal array to train PSO-BPN for further increasing the prediction accuracy.

Table 3. Artillery firing data

No.	Variables									Range (m)
	Following Wind Direction	Cross Wind Direction	Angle of Departure (^{o})	Muzzle Velocity (m/s)	Relative Humidity (%)	Air Pressure (mba)	Air Temperature (^{o}C)	Following Wind Velocity (m/s)	Cross Wind Velocity (m/s)	
1	Upwind	Upwind	1	1000	50	1002	5	1.7	1.7	2498
2	Upwind	Upwind	15	1005	75	1010	22	4	4	9334
3	Upwind	Upwind	30	1010	100	1019	39	6.3	6.3	12108
4	Upwind	Upwind	1	1000	50	1002	22	4	4	2528
5	Upwind	Upwind	15	1005	75	1010	39	6.3	6.3	9677
6	Upwind	Upwind	30	1010	100	1019	5	1.7	1.7	11246
7	Upwind	Upwind	1	1000	75	1019	5	4	6.3	2483
8	Upwind	Upwind	15	1005	100	1002	22	6.3	1.7	9341
9	Upwind	Upwind	30	1010	50	1010	39	1.7	4	12272
10	Upwind	Downwind	1	1000	100	1010	5	6.3	4	2488
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
80	Upwind	Downwind	30	1008	95	1015	17	3	6	11544
81	Upwind	Upwind	21	1001	88	1006	26	6	1.3	10561
82	Downwind	Upwind	5	1006	63	1011	21	2.8	2	6089

Table 4. Comparison of training time and prediction accuracy

NO.	Training Time (second)			MAPE (%)		
	O.A. PSOBPN	O.A. BPN	BPN	O.A. PSOBPN	O.A. BPN	BPN
1	20.456171	35.817796	123.519465	1.067	1.188	3.576
2	11.638934	36.284725	116.086414	1.207	1.308	3.600
3	25.207458	52.735658	102.330636	1.369	1.374	3.377
4	18.281387	30.371092	175.931698	1.315	1.239	3.687
5	17.647584	46.185630	366.937557	1.252	1.258	3.162

Note:
1. Software: Matlab 2010
2. Hardware: Intel(R) Core(TM) i5-2450M CPU@ 2.50GHZ, RAM 8.00 GB

Table 5. Error comparison of prediction and truth

No.	True Range (m)	O. A. PSO-BPN		O. A. BPN		BPN	
		Prediction (m)	Error (%)	Prediction (m)	Error (%)	Prediction (m)	Error (%)
1	8165	8177	0.145	8225	0.732	8267	1.252
2	9243	9475	2.506	9464	2.391	9530	3.103
3	8208	8154	0.663	8195	0.161	8409	2.444
4	6043	6043	0.007	6046	0.053	6604	9.287
5	10445	10309	1.306	10304	1.350	10255	1.817
6	10277	10045	2.258	10019	2.509	10030	2.403
7	10836	10729	0.984	10709	1.168	10578	2.383
8	11544	11227	2.748	11236	2.666	11045	4.324
9	10561	10659	0.930	10691	1.227	10389	1.632
10	6089	6100	0.184	6109	0.335	6465	6.169
MAPE (%)		1.173		1.259		3.481	

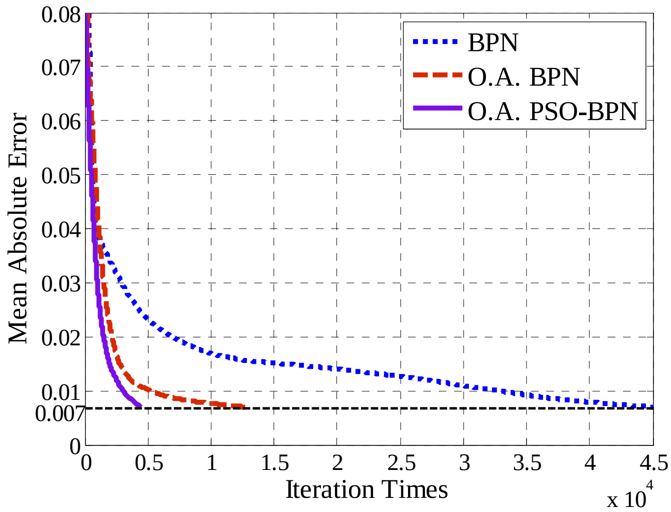

Fig. 3. Comparison of convergence process.

REFERENCES

Ala, B., Mounir, S., and Farhat, F. (2013). Fingerprint Verification Based on Back Propagation Neural Network. *Journal of Control Engineering and Applied Informatics.* 15 (3), 53-60.

Ahmed, H., Ahmed, Z. D., and Attia, D. (2013). Fault Detection and Isolation in Industrial Control Valve Based on Artificial Neural Networks Diagnosis. *Journal of Control Engineering and Applied Informatics.* 15 (3), 61-69.

Amit, K. S., Barjeev, T., and Vishal, K. (2013). Application of Neural Network Based Control Strategies to Binary Distillation Column. *Journal of Control Engineering and Applied Informatics.* 15 (4), 47-57.

Asanga, R., Saman, K. H., and Harry, C. W. (2004). Self-Organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients. *IEEE Transactions on Evolutionary Computation.* 8 (3), 240-255.

Chang, Y. J. and Tsai, J. J. (2008). Process Optimization Based on Neural Network Model and Orthogonal Arrays. *IEEE Congress on Evolution Computation*. 3486-3490.

Chang, Y. P., Low, C., and Wu, C. J. (2007). Optimal Design of Discrete-Value Passive Harmonic Filter Using Sequential Neural-Network Approximation and Orthogonal Array. *IEEE Transactions on Power Delivery*. 22 (3), 1813-1821.

Chang, C. G., Wang, D. W., Liu, Y. C., and Qi, B. K. (2007). Application of Particle Swarm Optimization Based BP Neural Network on Engineering Project Risk Evaluating'', *Third International Conference on Natural Computation.* 1, 750-754.

Dedy, P., Herman, A., and Mohd. F. R., (2008). The Taguchi-Neural Network Approach to Forecast Electricity Consumption. *IEEE Electrical and Computer Engineering.* 1941-1944.

Eberhart, R. C. and Kennedy, J. (1995, b). A New Optimizer Using Particle Swarm Theory. *Proc. Sixth International Symposium on Micro Machine and Human Science*. Nagoya, Japan, 39-43.

Fei, W. C. and Ing, J. J. (2008). The Impact Point and Ricochet Predictions of Cannon Shot. *Journal of Chung Cheng Institute of Technology*. 37 (1), 203-218.

Gross, B. (1988). Neural Network and Neural Intelligence. *Massachusettes institute of technology MIT Press*. 21-59.

Kennedy, J. and Eberhart, R. C. (1999). Empirical Study of Particle Swarm Optimization. Proceedings of the 1999 Congress on Evolutionary Computation. 3, 1945-1950.

Kennedy, J. and Eberhart, R. C. (1995, a). Particle swarm optimization. *Proc. IEEE International Conference on Neural Networks*. 4, 1942-1948.

Pian, Z., Li, S., Zhang, H., and Zhang, N. (2012). The Application of The PSO Based BP Network in Short-Term Load Forecasting. *Physics Procedia.* 24, 626-632.

Peters, S. and Koenig, A. (2007). A Hybrid Texture Analysis System Based on Non-Linear & Oriented Kernels, particle swarm optimization, and KNN vs. Support Vector Machines. *Neural Network World*. 507-527.

Su, M. C. (2007). The Study to Utilize The Neural Network to Predict The Artillery Range. *M.S. Thesis*. Chung Cheng Institute of Technology, National Defense University, Taiwan.

Sherif, Y. S. (1985). Modeling The Fire Tables of Field Artillery-Cannon 105mm Howitzer. *Microelectronic Reliability.* 25 (1), 41-53.

Wang, G. J., Tsai, J. C., Tseng, P. C., and Chen, T. C. (1998). Neural-Taguchi Method for robust design analysis. *Journal of the Chinese Society of Mechanical Engineers.* 19 (2), 223-230.

Yongzhong, L., David, L., Jingyu, Z., and Dahai L. (2011). An Evolutionary Performance-Enhanced PSO Approach by Using a BP Neural-Learning-Based PID Controller. *Connection Science.* 23 (1), 45-63.

Zhang, L., Ma, J., and Wang, Y., (2011). PSO-BP Neural Network in Reservoir Parameter Dynamic Prediction. *Seventh International Conference on Computational Intelligence and Security.* 123-126.