Damage Effect Assessment of Battlefield Target Based on Multiple Neural Network Fusion Algorithm

ZHI Jun
Engineering Institute of Engineering Corps
PLA University of Science and Technology, Nanjing, China
Zhenjiang Watercraft College of PLA, Zhenjiang,China
njzhijun@163.com

LIU Jian-yong and FU Cheng-qun
Engineering Institute of Engineering Corps
PLA University of Science and Technology, Nanjing, China
zhijun49717322@hotmail.com

Abstract—**The paper proposes an algorithm based on particle swarm optimization (PSO) and intuitionistic fuzzy sets theory. It fuses multiple different neural networks and applies it to the comprehensive assessment of battlefield target damage effect. It adopts PSO algorithm to improve and optimize multiple neural networks of different structure, then confirms the weights of different neural networks, and synthesizes their assessment results as the final output result according to the weight. Apply the algorithm to instance simulation, the result shows its validity and rationality.**

Index Terms—**damage effect assessment, PSO, intuitionistic fuzzy sets theory, multiple neural network**

I. INTRODUCTION

The model of the intuitionistic fuzzy sets is usually applied to decision-making problems. The characteristics of decision-making problems are as follows. It has multiple attributes, different attribute decision-making schemes have different good or bad degree, the policymaker has a partiality for decision scheme subjectively too. The intuitionistic fuzzy sets is to set up the intuitionistic judgment matrix and intuitionistic fuzzy decision-making matrix according to the characteristic of decision-making problem, then calculate the weight of the scheme or the index according to certain method, finally arrange the decision scheme in an order and select the superior.

Manuscript received October.15, 2009; revised October 30, 2009.
This work was supported in part by PLA University of Science and Technology under Grant 200803GB01

F. A. ZHI Jun is with Engineering Institute of Engineering Corps, PLA University of Science and Technology, Nanjing, 210007, China. Zhenjiang Watercraft College, Zhenjiang, 212003, China (njzhijun@163.com).
S. B. LIU Jian-yong is with Engineering Institute of Engineering Corps, PLA University of Science and Technology Nanjing, 210007, China. He is now a professor in operation research.
T. C. FU Cheng-qun is with Engineering Institute of Engineering Corps, PLA University of Science and Technology Nanjing, 210007, China.

While regarding neural network as an assessment method, correspondent to different samples which wait to be assessed, the good or bad contrast between different neural networks with different structures and training and optimization often present differences. So try to divide training samples into groups according to certain method while using training samples to train different neural networks. In this way, while regarding assessment problems as above-mentioned decision-making problems, different neural networks can be regarded as different "decision schemes", the samples divided into groups can be regarded as "attribute" of the decision-making problem. Namely we can consider using intuitionistic fuzzy sets theory to fuse multiple neural networks to improve the assessment accuracy.

II. PSO ALGORITHM OPTIMIZE NN

A .Particle Swarm Initialization According to Internal

Confirm the particles' positions according to different internals, set the position of *No.i* is $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$. If the number of particles is *m*, we can divide the solution space into *m* different regions, each region produces a particle, so the generation area of *No.i* particle is as follows.

$$X_{i,k} \in [a_k + f_k(b_k - a_k), a_k + f_{k+1}(b_k - a_k)] \quad k = 1,2,\ldots,D$$

$$\text{Where} \begin{cases} f_k = (i-1) \bmod C^{D-k} \\ f_D = (i-1) - \sum_{j=1}^{D-1} f_j C^{D-j} \\ C = \sqrt[D]{m} \end{cases} \quad (1)$$

"mod" expresses to fetch the arithmetical compliment, then we can generate the initial position of *No.i* according to formula (2).

$$X_{i,k} = a_k + f_k(b_k - a_k) + \frac{1}{\sqrt[D]{m}}(b_k - a_k)r \quad (2)$$

Where *r* is a random number between [0,1].

doi:10.4304/jcp.5.10.1590-1596

B. Adaptive Inertia Factors

At the later stage of algorithm iteration, most particles have already trended to approach the optimum position. If the inertia factor is still very great at this moment, it is very apt to deviate from the optimum position and very difficult to orient effectively. So consider making the inertia of the particle smaller when the particle is close to the optimum position. In this way the localization of the particle is the higher in precision, it can be close to the optimum position more effectively. The concrete realization method is as follows.

$$w = \frac{\sum_{j=1}^{D}(X_{gj} - X_j)}{\sum_{j=1}^{D} X_{gj}} \times w_0 \qquad (3)$$

Where w_0 is inertia factor, we usually fetch around 0.8, X_{gj} is the value of optimum position vector *No.j* dimension at present. X_j is the value of present position vector *No.j* dimension.

C. Initial Weight and Threshold Value Optimization

Whether the choice of initial weight and threshold value is rational relates to the final iterative effect. When there are several nodes in the network, we can find cooperatively ideal initial value through selecting experiment with groups of different weight and threshold value. But when there are more nodes in the network, the selection of weight and threshold value will meet the difficult problem similar to "combined explode". It is very difficult to solve the problem through the traditional experiment method. So we adopt PSO algorithm to optimize the initial weight and threshold value of neural network.

Mapping the weight and threshold value of network as a group of particles, namely one particle is a solution for a group of initial value.

The coding of particles adopts real value coding; the fitness function of each particle individual adopts the reciprocal of error function. So the larger the value of fitness function is, the finer the performance of the particle is.

III. FUSION ALGORITHM BASED ON INTUITIONISTIC FUZZY SETS and MULTIPLE NEURAL NETWORK

A. Intuitionistic Fuzzy Sets

Set X as a none empty aggregate, so we call F= $\{\langle x, \mu_F(x), v_F(x)\rangle \mid x \in X\}$ for intuitionistic fuzzy sets, $\mu_F(x)$ and $v_F(x)$ are respectively the element membership and non-membership of x belongs to X. $\mu_F \to [0,1], v_F \to [0,1]$, $0 \le \mu_F(x) + v_F(x) \le 1$, $\forall\ x \in X$.

Usually we take the general form of intuitionistic fuzzy sets abridged for $\alpha = (\mu_\alpha, v_\alpha)$, $0 \le \mu_\alpha + v_\alpha \le 1$. Its value can be calculated through the function $s(\alpha) = \mu_\alpha - v_\alpha$, $s(\alpha) \in$ [-1,1]. The score $s(\alpha)$ can be an important index to measure the intuitionistic fuzzy sets α .

B. IntuitionisticJudgment Matrix

When multiple neural networks to be carry on the evaluation, $Y = \{y_1, y_2, \cdots, y_n\}$ is a collection for neural network, through the comparison between *n* neural networks, we can construct a judgment matrix $B = (b_{ij})_{n\times n}$, $b_{ij} = (\mu_{ij}, v_{ij})$ ($i, j = 1,2,\cdots,n$). μ_{ij} shows the relative credibility of neural network x_i compared to x_j. v_{ij} shows the relative credibility of neural network x_j compared to x_i . Calculation formula of the relative credibility is as follows.

$$\mu_{ij} = \frac{E_j}{E_i}\ ,\ v_{ij} = \frac{E_i}{E_j}\ ,\ i, j = 1,2,\cdots n \qquad (4)$$

In formula (4), E_i and E_j respectively shows error value of the neural network *i* and *j* after going on learning and training to the overall sample. Thus we have constructed an intuitionistic judgment matrix of multiple neural networks based on overall samples.

C. Intuitionistic Fuzzy Decision-making Matrix

We can construct an intuitionistic fuzzy decision-making matrix similar to the intuitionistic judgment matrix according to the satisfactory degree of the decision scheme to the decision problem attributes. $A = \{A_1, A_2, \cdots, A_n\}$ shows as a collection for neural network, $G = \{G_1, G_2, \cdots, G_m\}$ is a collection of grouped samples, $w = (w_1, w_2, \cdots, w_m)^T \in H$ is the weight vector quantity of the grouped samples. In the decision-making matrix $R = (r_{ij})_{n\times m}$, $r_{ij} = (\alpha_{ij}, \beta_{ij})$, α_{ij} denotes the credibility of the neural network A_i to grouped samples G_j , β_{ij} denotes the none credibility of the neural network A_i to grouped samples G_j . $0 \le \alpha_{ij} + \beta_{ij} \le 1$, $i = 1,2,\cdots,m; j = 1,2,\cdots,n$. Calculation formula of credibility and none credibility are as follows.

$$\theta_i = \frac{C - E_i}{\sum_{k=1}^{n}(C - E_k)}, \phi_i = \frac{E_i}{\sum_{k=1}^{n} E_k}\ ,\ i = 1,2,\cdots n \qquad (5)$$

In formula (5), E_i shows the error value of the neural network *i* after going on learning and training to the grouped samples. *C* is a positive constant, we usually get one. Thus we have constructed an intuitionistic fuzzy decision-making matrix of multiple neural networks based on grouped samples.

We can utilize score function to calculate and get the score matrix of fuzzy decision-making matrix, that is $S=(s(r_{ij}))_{n\times m}$, $s(r_{ij})=s(\alpha_{ij},\beta_{ij})=\alpha_{ij}-\beta_{ij}$, $s(r_{ij})\in[-1,1], i=1,2,\cdots,n; j=1,2,\cdots,m$.

Then use formula (6) to get $\bar{S}=(\bar{s}(r_{ij}))_{n\times m}$ utilizing $S=(s(r_{ij}))_{n\times m}$:

$$\bar{s}(r_{ij})=\frac{s(r_{ij})-\min_i\{s(r_{ij})\}}{\max_i\{s(r_{ij})\}-\min_i\{s(r_{ij})\}} \quad (6)$$

$$i=1,2,\cdots,n; j=1,2,\cdots,m$$

According to the standardized score matrix $\bar{S}=(\bar{s}(r_{ij}))_{n\times m}$, the synthetical score of each neural network can be shown as follows.

$$\bar{s}(r_i)=\sum_{j=1}^{m}w_j\bar{s}(r_{ij}) \quad (7)$$

$$i=1,2,\cdots,n$$

But in formula (7), the weight information is completely unknown, so we need to set up certain model to get the solution of attribute weight.

D. Additive Consistent Linear Programming Model

In order to try to get the attribute weight, we introduce the additive consistent linear programming model to calculate. Combined with the theory of intuitionistic fuzzy sets, we provide the following definition of additive consistent linear intuitionistic judgment matrix.

Set $B=(b_{ij})_{n\times n}$ as intuitionistic judgment matrix, $b_{ij}=[\mu_{ij},1-v_{ij}]$ ($i,j=1,2,\cdots,n$). If exist the vector $w=(w_1,w_2,\cdots,w_n)^T$ to get $\mu_{ij}\le 0.5(w_i-w_j+1)\le 1-v_{ij}$, $i,j=1,2,\cdots,n$. $w_j\ge 0$, $\sum_{j=1}^{n}w_j=1$, we call B for additive consistent judgment matrix.

In order to make the decision information identical, we use the synthetical score of all the neural network A_i ($i=1,2,\cdots n$) to construct additive consistent linear complementary judgment matrix $\bar{B}=(\bar{b}_{ij})_{n\times n}, i,j=1,2,\cdots,n$

In order to get the attribute weight vector $w=(w_1,w_2,\cdots;w_n)^T$, we establish the following linear programming model.

$$w_k^-=\min w_k \quad (8)$$

$$s.t.\begin{cases}0.5(\sum_{k=1}^{m}w_k(\bar{s}(r_{ik})-\bar{s}(r_{jk}))+1)+d_{ij}^-\ge\mu_{ij}\\ 0.5(\sum_{k=1}^{m}w_k(\bar{s}(r_{ik})-\bar{s}(r_{jk}))+1)-d_{ij}^+\le 1-v_{ij}\\ w=(w_1,w_2,\ldots,w_n)^T\in H, w_i\ge 0,\sum_{i=1}^{n}w_i=1\end{cases}$$

$$w_k^+=\max w_k \quad (9)$$

$$s.t.\begin{cases}0.5(\sum_{k=1}^{m}w_k(\bar{s}(r_{ik})-\bar{s}(r_{jk}))+1)+d_{ij}^-\ge\mu_{ij}\\ 0.5(\sum_{k=1}^{m}w_k(\bar{s}(r_{ik})-\bar{s}(r_{jk}))+1)-d_{ij}^+\le 1-v_{ij}\\ w=(w_1,w_2,\ldots,w_n)^T\in H, w_i\ge 0,\sum_{i=1}^{n}w_i=1\end{cases}$$

In formula (8) and (9), d_{ij}^- and d_{ij}^+ are not minus constant, they are introduced deviation variable while the additive consistent linear complementary judgment matrix $\bar{B}$ disaccords with the intuitionistic judgment matrix constructed by the comparison between n neural networks. When matrix $\bar{B}$ and B go all the way, d_{ij}^- and d_{ij}^+ are expressed as zero. We can judge whether the matrix $\bar{B}$ and B go all the way using formula (10).

$$\mu_{ij}\le 0.5(\sum_{k=1}^{m}w_k(\bar{s}(r_{ik})-\bar{s}(r_{jk}))+1)\le 1-v_{ij} \quad (10)$$

The weight vector quantity that satisfies the condition is more than one. So every weight w_k belongs to certain zone. Through solve the model (8) and (9), we can get the collection of attribute weight vector:

$$\Delta_1=\{w=(w_1,w_2,\cdots,w_m)^T$$

$$w_k\in[w_k^-,w_k^+], w_k\ge 0, k=1,2,\cdots,m,\sum_{k=1}^{m}w_k=1\}$$

IV. FUSION ALGORITHM DESIGN

Step1: Particle swarm optimization: According to the structure of neural network, initialize the initial position, speed, inertia weight of the particle etc. Stipulate the scale of the population, convergence condition etc. Calculate the fitness of particle according to training samples and begin to iterate, upgrade the speed and position of the particle, output the result finally, and confirm the initial weight and threshold value of the neural network.

Step2: Set up the following linear programming model to get the weight vector quantity of optimum attribute according to the obtained interval vector quantity of attribute weight:

$$\varphi=Max\sum_{i=1}^{n}\sum_{j=1}^{m}(1-\beta_{ij}-\alpha_{ij})w_j \quad (11)$$

$$s.t. \quad w=(w_1,w_2,\ldots,w_n)^T\in\Delta$$

The obtained weight vector quantity of optimum attribute is as follows.

$$w^*=(w_1^*,w_2^*,\cdots,w_n^*)^T$$

Step3: Calculate the comprehensive score of each neural network

$$z_i(w^*) = [z_i^-(w^*), z_i^+(w^*)](i = 1,2,\cdots,m) \quad (12)$$

In formula (12)

$$z_i^-(w^*) = \sum_{j=1}^{m} w_j^* \alpha_{ij} \quad z_i^+(w^*) = \sum_{j=1}^{m} w_j^*(1-\beta_{ij})$$

Step4: Construct the credibility matrix $P = (p_{ij})_{n\times n}$ based on the comparison between the comprehensive score $z_i(w^*)(i = 1,2,\cdots,n)$ of every neural network.

$$p_{ij} = p(z_i(w^*) \ge z_j(w^*))$$
$$= \max\left\{1 - \max\left(\frac{z_j^+(w^*) - z_i^-(w^*)}{z_i^+(w^*) - z_i^-(w^*) + z_j^+(w^*) - z_j^-(w^*)}, 0\right), 0\right\}$$

Step5: Obtain the weight vector quantity of neural network according to the following formula:

$$\omega = (\omega_1, \omega_2, \cdots, \omega_n) \quad (14)$$

$$\omega_i = \frac{1}{n(n-1)}\left(\sum_{j=1}^{n} p_{ij} + \frac{n}{2} - 1\right)$$

Step6: Finally fuse the assessed output vector $O = (o_1, o_2, \cdots, o_n)$ of *n* neural networks.

$$R = \sum_{i=1}^{n} o_i * \omega_i \quad (15)$$

R is the weighting synthetical outputs of the multiple neural networks.

V. INSTANCE ANALYSES

We use 122mm shrapnel and 152mm shrapnel as the kind of artillery weapon, and regard 122 howitzers, 152 cannons and 130 cannons as the target of battlefield. We set up the damage assessment model of the artillery to the battlefield target, and use the fuzzy neural network to carry on the damage effect assessment.

A. Group of Training Samples

Table 1 is the damage assessment standard created by experts, corresponding different destruction degree of targets to set up different blocks of damage.

TABLE 1

GRADE STANDARD OF DAMAGE DEGREE

DD	A	B	C	D	E.	F	G
GD	0~ 0.05	0.05~ 0.2	0.2~ 0.4	0.4~ 0.6	0.6~ 0.75	0.75~ 0.95	0.95 ~1

DD is the logogram of damage degree; GD is the logogram of grade Standard. A denotes no damage, B denotes slight damage, C denotes low-grade damage, D denotes medium-sized damage, E denotes little serious damage, F denotes serious damage, G denotes scrap.

There are sixteen training samples sifted for the neural network train in table 2 to table 5. Every four training samples will be divided into one group and we get a, b, c, d four groups. There are four test samples in table 6. The concrete explanations for the tables are as follows.

For X_1, 0 denotes 122mm shrapnel, 1 denotes 152mm shrapnel;

For X_2, the unit is m;

For X_3, the value range is $[-\pi/2, \pi/2]$;

For X_4, the value range is $[0, \pi/2]$;

For X_5, the value range is $[0, 2\pi]$;

For X_6, 0 denotes 122mm howitzers, 1 denotes 152mms cannons, 2 denote 130mm cannons;

For X_7, 0 denotes that the target has no blindage, 1 denotes half blindage, 2 denote simple blindage, 3 denote firm blindage.

TABLE 2

NEURAL NETWORK TRAINING GROUP SAMPLES A

N	X_1	X_2	X_3	X_4	X_5	X_6	X_7	score	DD
1	0	3.6	0.14	0.36	0.69	1	0	0.85	F
2	1	10.1	-0.1	0.35	5.26	2	1	0.12	B
3	1	22	-0.02	0.35	4.8	2	1	0.07	B
4	0	8.2	0.12	0.37	0.9	0	2	0.3	C

TABLE 3

NEURAL NETWORK TRAINING GROUP SAMPLES B

N	X1	X2	X3	X4	X5	X6	X7	score	DD
5	1	6.6	0.13	0.35	5.49	2	1	0.18	B
6	0	7.8	0.06	0.36	5.17	1	0	0.15	B
7	1	8.6	0.05	0.34	4.54	2	1	0.5	D
8	1	14.3	-0.02	0.34	1.47	2	1	0.42	D

TABLE 4

NEURAL NETWORK TRAINING GROUP SAMPLES C

N	X1	X2	X3	X4	X5	X6	X7	score	DD
9	1	8.3	0.12	0.35	1.19	2	1	0.46	D
10	1	7.8	0.05	0.34	4.23	2	1	0.7	E
11	0	4	0.15	0.37	1.04	1	0	0.37	C
12	1	13.8	-0.04	0.35	2.04	2	1	0.06	B

TABLE 5

NEURAL NETWORK TRAINING GROUP SAMPLES D

N	X1	X2	X3	X4	X5	X6	X7	score	DD
13	1	13.4	0.01	0.34	1.27	2	1	0.42	D
14	1	5.3	-0.01	0.35	2.06	2	1	0	A
15	1	10.2	0.11	0.36	0.99	1	0	0.06	C
16	1	15.4	-0.07	0.34	3.8	2	1	0.07	B

TABLE 6

NEURAL NETWORK TEST SAMPLES

N	X1	X2	X3	X4	X5	X6	X7	score	DD
1	1	3	0.25	0.35	5.54	2	1	0.93	F
2	1	7.5	0.12	0.35	0.73	2	1	0.44	D
3	0	15	-0.18	0.38	2.76	1	0	0	A
4	1	9.6	-0.003	0.35	5.12	2	1	0.35	C

B. Error Count of Neural Network Training

Adopt three kinds of neural networks with different structures, which are A_1, A_2, A_3.The numbers of the latent layer of neurons are respectively set as 10 , 12 , 14.

We can get four collections G1, G2, G3 and G4 from the training group samples a, b, c and d. Collect G includes all the group samples a, b, c, d. The sample collection G1 , G2 , G3 , G4 and G are trained respectively by the neural network A_1, A_2, A_3. The training course is as Fig.1 to Fig.3 show. The dotted lines in the figure denote the result of neural network iterating 50 times which has not been optimized by PSO algorithm. The full lines in the figure denote the result of neural network iterating 50 times which has been optimized by PSO algorithm.

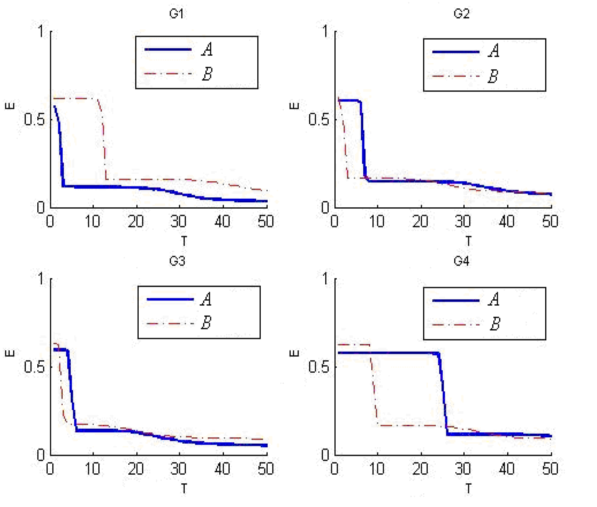

Figure 1 NN A_1 training result

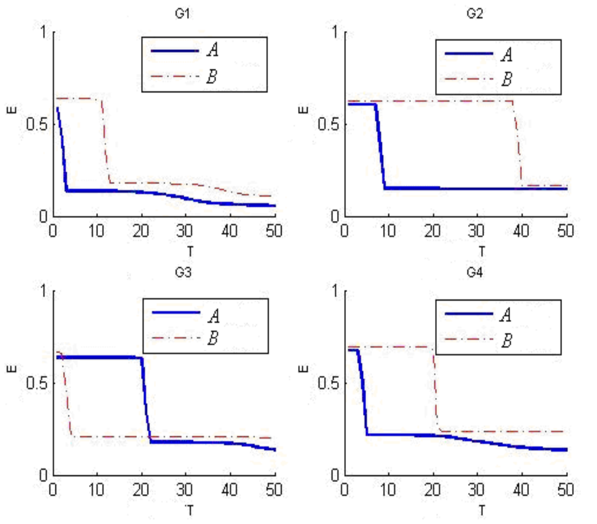

Figure 2 NN A_2 training result

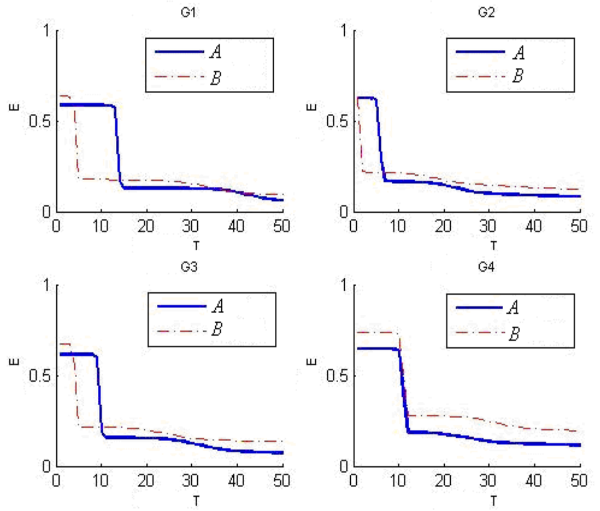

Figure 3 NN A_3 training result

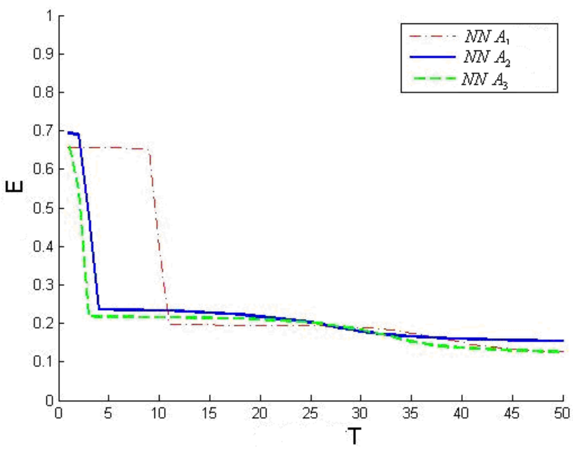

Figure 4 Overall sample collection training result

We can see from Fig.1 to Fig.3, on one hand the neural network with PSO demonstrates obvious performance during the training. On the other hand to different sample collections, different neural networks demonstrate different performance, “credibility” is not the same. Fig.4 is the training iterative course of three kinds of neural networks optimized by PSO algorithm to overall sample *G*.

The sample collection G1, G2, G3, G4 and G are trained respectively ten times by the neural network A_1, A_2, A_3.We fetch the average error as the calculation foundation for intuitionistic judgment matrix and intuitionistic fuzzy decision-making matrix. Table.7 shows the result of the average error calculation.

TABLE 7

RESULT OF AVERAGE ERROR CALCULATION

	E1	*E2*	*E3*	*E4*	*E*
A1	0.05	0.06	0.09	0.13	0.03
A2	0.02	0.06	0.05	0.03	0.05
A3	0.04	0.08	0.08	0.11	0.04

C. Ascertain the weight of different neural networks

We set up the intuitionistic fuzzy decision-making matrix *R* according to the average error E1, E2, E3, E4. It shows in table 8.

TABLE 8

INTUITIONISTIC FUZZY DECISION MATRIX R

	G1	*G2*	*G3*	*G4*
A1	(0.17,0.45)	(0.34,0.30)	(0.33,0.41)	(0.32,0.48)
A2	(0.69,0.18)	(0.34,0.30)	(0.34,0.23)	(0.10,0.11)
A3	(0.14,0.37)	(0.32,0.40)	(0.33,0.36)	(0.58,0.41)

According to the average error E from the neural network A1, A2 and A3 to the overall sample G, we set up the intuitionistic judgment matrix B as table.9 shows after normalization.

TABLE 9

INTUITIONISTIC JUDGEMENT MATRIX B

	A1	A2	A3
A1	(0.5,0.5)	(0.3,0.7)	(0.4,0.6)
A2	(0.7,0.3)	(0.5,0.5)	(0.8,0.2)
A3	(0.6,0.4)	(0.2,0.8)	(0.5,0.5)

Calculate the score matrix S according to the intuitionistic fuzzy decision-making matrix R

TABLE 10

SCORE MATRIX S

	G_1	G_2	G_3	G_4
A_1	-0.28	0.04	-0.07	-0.16
A_2	0.51	0.04	0.11	-0.01
A_3	-0.23	-0.08	-0.03	0.17

Matrix S will be turned into the standardized matrix $\overline{S}$

TABLE 11

STANDARDIZED MATRIX $\overline{S}$

	G_1	G_2	G_3	G_4
A_1	0	1	0.75	0.375
A_2	1	0.096	0.231	0
A_3	0	0.375	0.5	1

We can get the optimum attribute weight vector quantities of G1, G2, G3 and G4 according to the linear programming model.

$$w^* = (0.3262,0.1907,0.3805,0.1125)$$

Then we can get the comprehensive score of the neural network A1, A2, A3 according to formula (10) and (11).

$$z_1 = [0.4957, 0.6768]$$
$$z_2 = [0.5815, 0.8153]$$
$$z_3 = [0.6198, 0.7677]$$

Then we can set up the credibility matrix according to the comparison between the synthetical scores.

TABLE 12

CREDIBILITY MATRIX P

	A_1	A_2	A_3
A_1	0.5	0.1825	0.5155
A_2	0.8576	0.5	0.7124
A_3	0.5792	0.3022	0.5

We can get the weight vector quantities of A1，A2,A3 utilizing formula(14).

$$\omega = (0.2684, 0.4458, 0.3149)$$

D. Assess the test samples

Then we can get the weighting collection using the weight vector to the assessment value of three kinds of neural networks. The assessment value got finally is that the fuzzy artificial neural network fusion algorithm to the assessment of test samples. The assessed results and analyses shows in the Table 13.

We can find that when we use the fusion algorithm to assess, the error is lower compared to the single neural network except for the assessment for test samples 3. It signifies that the fusion algorithm is effective.

V.CONCLUSIONS

The paper combines the intuitionistic fuzzy sets theory and PSO algorithm with the neural network, and applies it to the comprehensive evaluation of target damage effect in the battlefield. Through the instance analysis，we have verified its validity and rationality.

TABLE 13

ASSESSED RESULT OF TEST ASMPLES

	Test samples 1			Test samples 2		
	Assessing value	Actual value	Error	Assessing value	Actual value	Error
A1	0.89	0.93	0.04	0.41	0.44	0.03
A2	0.95	0.93	0.02	0.39	0.44	0.05
A3	0.91	0.93	0.02	0.47	0.44	0.03
A4	0.94	0.93	0.01	0.43	0.44	0.01
	Test samples 3			Test samples 4		
A1	0	0	0	0.40	0.35	0.05
A2	0.07	0	0.07	0.37	0.35	0.02
A3	0.15	0	0.15	0.33	0.35	0.02
A4	0.06	0	0.06	0.36	0.35	0.01

REFERENCES

[1] FAN Wei-ming. Study on the neural network based on the genetic algorithm [J]. Journal of Taiyuan university, 2004, 3(4): 14-17.

[2] WANG Feng-qing, GAO Yin. Optimization of the neural net work based on the genetic algorithm[J]. Journal of Yanshan University, 2001, 25(3): 234-238.

[3] Hong D H, Choi C H. Multicriteria fuzzy decision-making problems based on vague set theory [J]. Fuzzy Sets and Systems, 2000, 114(1):103-113.

[4] Mikhailov L.Deriving priorities from fuzzy pairwise comparison judgments [J].Fuzzy Sets and Systems, 2005, 152(3):475-498.

[5] Xu Z S, chen J. An iterative method for fuzzy multiple attributes group decision making [J].Information Sciences, 2007, 177(1):248-263.

[6] Xu Zeshui. Uncertain Multiple Attribute Decision Making: Methods and Applications [M].Beijing: Tsinghua University Press, 2004.