Available online at www.sciencedirect.com
ScienceDirect
Applied Mathematics and Computation 195 (2008) 66–75
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

A dynamic all parameters adaptive BP neural networks model and its application on oil reservoir prediction

Shiwei Yu *, Kejun Zhu, Fengqin Diao

School of Management, China University of Geosciences, Wuhan, Hubei 430074, PR China

Abstract

In this paper, a dynamic all parameters adaptive BP neural networks model is proposed by fusing genetic algorithms (GAs), simulated annealing (SA) and error back propagation neural network (BPNN) to offset the demerits of one paradigm by the merits of another. Adopting multi-encoding, the model can optimize the input nodes, hidden nodes, transfer function, weights and bias of BP networks dynamically and adaptively. Under accurate premise, the simple architecture (less input and hidden nodes) of network model is constructed in order to improve networks' adaptation and generalization ability, and to greatly reduce the subjective choice of structural parameters. The results of application on oil reservoir prediction show that the proposed model with comparatively simple structure can meet the precision request and enhance the generalization ability.

Keywords: Dynamic; All parameters; Adaptive; Genetic algorithms; BP neural network; Structure identification

1. Introduction

Artificial neural networks (ANNs), due to their excellent ability of non-linear mapping, generalization, self-organization and self-learning, have been proved to be of widespread utility in engineering and are steadily advancing into new areas [1–5]. The 'feed-forward, back propagation' neural network usually trained by back-propagation of errors (BPNN) is perhaps the most popular network architecture in use today [6]. Although encouraging results have been reported in which ANNs-based system outperformed widely-used well-established statistical methods, many inconsistent reports have been undermining the robustness of these findings. The reasons of these discrepancies are well-known problems characterized BPNN: (1) As the BP algorithm optimizes a target function by using the gradient descent method, the calculation may overflow or fluctuate between the optima; (2) BPNN is sensitive to the choice of topology and size. Different hidden nods and transfer function may lead to different results when BPNN is trained and tested on the same database; (3) The convergence of BPNN is also sensitive to the initial selection of weights. If the initial sets of

* Corresponding author.
E-mail address: ysw81993@sina.com.cn (S. Yu).

0096-3003/$ - see front matter
doi:10.1016/j.amc.2007.04.088

weights are not selected properly, the optimization solution could be trapped in a local optimum; (4) Generally, BP algorithms adjust weights and bias complying with a certain rule. It is impossible to adjust structure self-adaptively in a fixed topology.

Despite the increasing level of research activity the problem described has not yet been solved definitively. Different approaches can be categorized as follows: (1) empirical or statistical methods, that are used to study the effect of ANNs internal parameters and choose appropriate values for themselves based on the model's performance [7–10]. The most systematic and general of these methods utilize the principles from Taguchi's design of experiments [11]. The best combination of the number of hidden layers, hidden neurons, choice of input factors, training algorithm parameters, etc., can be identified by these methods even though they are mostly case-oriented. (2) Grey correlation analysis [12]. In the case of that grey correlation analysis is used to determine the number of the hidden nods in the BPNN's optimal network and to improve the network's performance. (3) Network growing/or pruning algorithms that, respectively, add and/or remove neurons from an initial architecture using a previously specified criterion to indicate how ANNs' performance is affected by changes [13–17]. The basic rules are that neurons are added when training is slow or when the mean squared error is larger than a specified value, and that neurons are removed when a change in a neuron's value does not correspond to a change in the network's response or when the weight values that are associated with this neuron remain constant for a large number of training epochs. Since both constructive and pruning algorithms are basically gradient descent methods, their convergence to the global minimum cannot be guaranteed. Therefore they can be trapped to a local minimum close to the point of the search space where the algorithm starts. (4) Evolutionary strategies that search over topology space by varying the number of hidden layers and hidden neurons through application of genetic operators and evaluation of the different architectures according to an objective function [17–21]. In recent years, the considerable progress has been made in last category. However, most of previous researchers hybridized two of evolutionary techniques such as GA–BP, SA–BP, GA–SA. Little has fused GAs, SA, and BP synchronously.

In this paper, a dynamic all parameters adaptive BP neural networks model is proposed by fusing genetic algorithms (GAs), simulated annealing (SA) and error back propagation neural network (BPNN) to offset demerits of one paradigm by merits of another. The algorithms whose every chromosome structures consist of both binary and real parts are briefly described as following: Firstly, the model optimizes input nodes, hidden nodes, transfer function, weights and bias of BP networks utilizing GA. Secondly, in order to get proper weights and bias of a fixed network structure, the real part of chromosome corresponding weights and bias value go to SA–BP hybrid algorithms with a certain probability. The better weights and bias can be got by using SA global search avoiding being trapped in a local minimum when adjusted by BP algorithm. Finally, set weights and bias obtained by SA–BP to corresponding chromosome, and then go to next generation operation. Under the premise of ensuring accuracy, the architecture of the network model is relatively simple (less input and hidden nodes) so that to improve adaptation and generalization ability of networks, and to greatly reduce subjective choice of structural parameters.

The paper is organized as follows. In Section 2, we describe design of the model in detail. The algorithm approach of adaptive optimize parameters of BPNN is described in Section 3. In Section 4, applications are presented and the performances of the various existing methods are compared. We conclude in Section 5.

2. Design of the model

GA is a global search procedure that searches from one population of points to another. As the algorithm continuously samples the parameter space, the search is directed toward the area of the best solution so far. This algorithm has been shown to perform exceedingly well in obtaining global solutions for difficult non-linear functions [4]. The application of the GA to one particularly complex non-linear function, the ANN, has also been shown to dominate other more commonly used search algorithms [19,20,22]. It has some main disadvantages, for example, slow convergence speed; premature convergence or stalling; the larger number of populations and individuals for some complex optimization problems. SA, as its name implies, exploits an analogy between the way in which a metal cools and freezes into a minimum energy crystalline structure (the annealing process) and the search for a minimum in a more general system. The algorithm is based upon that of Kirkpatrick et al. [23], which was originally proposed as a means of finding the equilibrium configu-

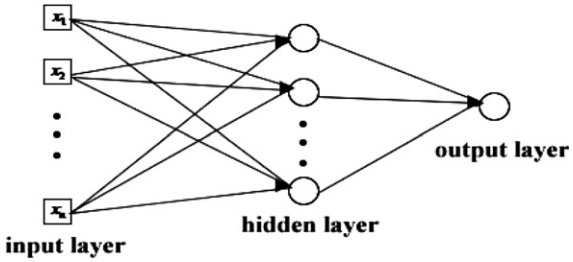

Fig. 1. A typical back-propagation ANNs model.

ration of a collection of atoms at a given temperature. SA's major advantage over other methods is the ability if avoiding becoming trapped at local minima. The algorithm employs a random search, which not only better but also worse neighboring solutions with a certain probability. SA is a powerful optimization technique to solve the combinatorial optimization problems, but it is computationally intensive, especially for more complicated problems [24]. BP algorithm which is mostly used to train networks is a gradient-based method; hence some inherent problems are frequently encountered in the use of this algorithm, e.g., very slow convergence speed in training, easily to get stuck in a local minimum, etc. [25]. To overcome these disadvantages, combining GA, SA and BP could achieve more effective results. This study employs genetic algorithm to carry out near-optimal neural network structure, and SA, BP to realize optimal weights and bias of the network.

The common three-layer feed-forward type of ANNs, as shown in Fig. 1, is considered. This is the type of network in which the units each perform a biased weighted sum of their inputs and pass this activation level through a transfer function to produce their output, and the units are arranged in a layered feed-forward topology.

To dynamic and adaptive optimize all parameters of BP neural networks by a hybrid algorithm of GA–SA–BP; first of all, design as following must be done in this section.

2.1. Chromosome multi-encoding

In order to optimize BP neural network structure, transfer function, weights and bias at the same time, we used GA with multi-encoding. Chromosome structures with six gene segments include both binary and real parts, as it is demonstrated in Fig. 2 and presented as chromsom $= [X^1, X^2, X^3, X^4, W^1, W^2]$. Neurons of network in the input and hidden layer, as well as the transfer function type of neurons are coded by the first part of chromosomes structure. If the value of gene X^1 and X^2, coded in binary system, is "1", the neuron is selected, on the contrary, if the value is "0", the neuron is unselected. For example, there are 10 inputting variables, the masking binary string encoded as [1 1 1 1 1 1 1 0 0 0] presents that the first seven inputs of feature work piece parts should be kept, and the last three inputs should be removed. X^3 and X^4 are the genes of determination the type of transfer function of hidden and output layer neurons respectively. For example, code binary string as 0 0 0, 0 0 1, 0 1 1 representing the transfer function of hanrdlim, log sig and tan sig respectively. Those

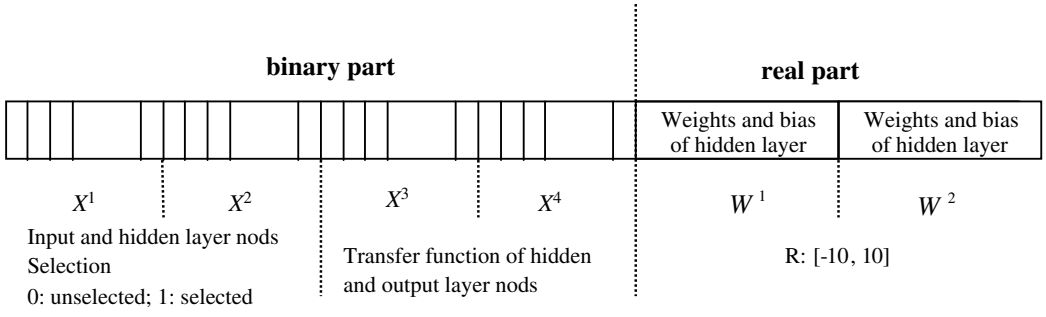

Fig. 2. GANN chromosome structure.

genes located in the real part, W^1 and W^2 are use to code the weights and bias parameters which may be in values of $[-10, 10]$.

2.2. Selection operator

Generally, selection operator of genetic algorithm is implemented by using roulette-wheel algorithm. The main defect of the roulette-wheel algorithm is that local optimal gene dominates the whole generation, while global optimum would likely be eliminated before emerging. Therefore, the roulette wheel with elitist selection method is considered as the selection mechanism in this proposed GA–SA–BP algorithm. This method including two phases, at first, the elitist strategy was adopted. The generation whose fitness is within top 10% will be directly copied to the new generation. Then the rest of chromosome is selected by roulette wheel.

2.3. Crossover operator

Crossover is a mechanism of randomly exchanging information between two chromosomes. Different crossovers mechanisms for binary and real part are implemented in this study. The two-point crossover mechanism is applied to the binary part of chromosome, which is demonstrated in Fig. 3.

The following crossover for real part: Let chrom1 and chrom2 are two chromosomes, and are two independently distributed random variables with range $[0, 1]$. We can get two new chromosome by Eq. (1):

$$\begin{cases} \text{chrom1}' = \text{chrom1} + (1 - r_1) \cdot (\text{chrom2} - \text{chrom1}), \\ \text{chrom2}' = \text{chrom2} + (1 - r_2) \cdot (\text{chrom1} - \text{chrom2}). \end{cases} \tag{1}$$

2.4. Mutation operator design

The purpose of mutation operation is to make genetic algorithm obtain local random research capability through varying certain genes of chromosome. If a chromosome is selected for mutation, one gene is randomly selected for changing their values. In the respect of binary representations the process is particularly simple: randomly choose a locus and switch its value, either $0 \mapsto 1$ or $1 \mapsto 0$. While, for the real part of the chromosome, let p is a parent, then the child chromosome is, $p\prime = p + \epsilon$ where is a distributed random variables with range $[0, 1]$.

2.5. Fitness function

A fitness value is given by Eq. (2):

$$\text{fitness} = \frac{1}{E + \alpha \cdot (\text{input\_no} + \text{hide\_nod})}, \tag{2}$$

where E is the mean squared error of neural network. α $(0 < \alpha < 0.01)$ is the affection coefficient of the number of input nods and hidden layer nods. input_no, hide_nod are the number of input nods and hidden layer nods, respectively. Furthermore, one input nod and one hidden layer nod must be selected at least in every chromosome, namely, input_no $\geqslant 1$, hide_nod $\geqslant 1$.

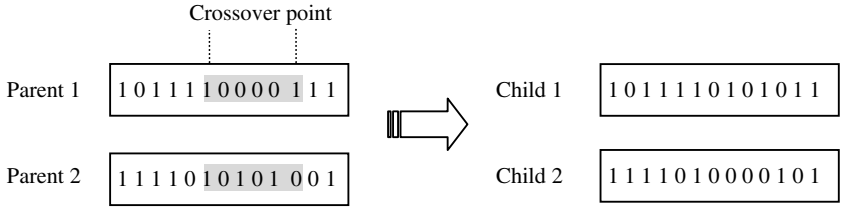

Fig. 3. Two-point crossover.

2.6. Adaptive probability of crossover and mutation

Traditional crossover and mutation operators are based on a randomization mechanism, i.e., generating a cut point, and determining the position of the bit shifted by mutation of the solution. But this is not the case in natural evaluation which is mimicked by the GA. Actually renewing the bits of the solution is dynamic or adaptive, but not random. The slightly modified adaptive probabilities of crossover and mutation given by Srinivas and Patnaik [26] are used in this study to choose the probability of mutation and crossover according to the fitness value of the solutions. The modified expression of p_c and p_m are as follows:

$$P_c = \begin{cases} (f_{max} - f')/(f_{max} - f_{ave}) & \text{if } f' \geqslant f_{avg}, \\ 1.0 & \text{if } f' < f_{ave}, \end{cases} \tag{3}$$

$$P_m = \begin{cases} 0.5 \cdot (f_{max} - f)/(f_{max} - f_{ave}) & \text{if } f \geqslant f_{ave}, \\ (f_{ave} - f)/(f_{ave} - f_{min}) & \text{if } f < f_{ave}. \end{cases} \tag{4}$$

Here, f is the fitness of an individual, f_{ave} the average fitness value of the population, and f_{max} and f_{min} the maximum and minimum fitness value of the population respectively. f' is the larger of the fitness values of the solutions to be crossed.

3. The algorithm approach of adaptive optimize parameters of BPNN

The algorithm approach of optimizing the network structure, transfer function and weights value using GA combining SA and BP include two phases: firstly, the model optimizes the input nodes, hidden nodes, transfer function, weights and bias of BP networks by utilizing GA. Secondly, in order to get proper weights and bias of a fixed network structure, the real part of chromosome's corresponding weights and bias value go to SA–BP hybrid algorithms with a certain probability. The better weights and bias can be got by using SA global search avoiding being trapped in a local minimum when adjusted by BP algorithm. The more details are described as follows:

(1) Standardize the training and testing data.
(2) Initialize prior parameters of the model: the size of population = pop_size, the max generation max gen.
(3) Generate initial population chromsom = $[X^1, X^2, X^3, X^4, W^1, W^2]$, in which X^1, X^2, X^3, X^4 are binary codes and W^1, W^2 are real number codes and the value of fitness is calculated.
(4) The operation of relative genetic operators.
(4.1) Select and reproduce population according to value of fitness.
(4.2) Calculate the self-adaptive crossover probability P_c, and crossover operation of different code genes.
(4.3) Calculate the self-adaptive crossover probability P_m, and mutation operation of different code gene.
(4.4) Estimate the number of 1's in the X^1 gene code, if the number is 0, one of the genes set to 1 randomly. The same principle is also used in the X^2, else go to (5).
(5) Utilize SA–BP to get the optimal network weights basing on the fixed input, structure, and transfer function, and let $p = 1$.
(5.1) If $p \leqslant$ pop_size, go to (5.2), else go to (6).
(5.2) chromsom$_p$ will be entered into SA–BP based on certain probability, which means if rand $< P_{SABP}$, go to (5.3), else $p = p + 1$, and go to (5.1).
(5.3) Set the initial annealing temperature t_0, final temperature t_s, annealing ratio α and length of Markov L. If $t \leqslant t_s$ the individual optimized by BP algorithm and can get chromsom$_p = [X_p^1, X_p^2, X_p^3, X_p^4, W_p^{1*}, W_p^{2*}]$, else let $p = p + 1$, go to step (5.2).
(5.4) Utilize SA to optimize the chromsom$_p = [X_p^1, X_p^2, X_p^3, X_p^4, W_p^{1*}, W_p^{2*}]$ globally, and get the individual optimized weights chromsom$_p = [X_p^1, X_p^2, X_p^3, X_p^4, W_p^{1**}, W_p^{2**}]$. Let $p = p + 1$ go to (5.1).
(6) Calculate fitness value of the population which optimized by GA–SA–BP from step (4) and (5), chromsom = $[X^1, X^2, X^3, X^4, W^{1**}, W^{2**}]$.
(7) Let gen = gen + 1, if gen $\leqslant \max gne$, go to next generation and implement a new GA–SA–BP operation from the Step (4), else stop the iteration, get the best individual best_chromsome,namely the best input number, hidden nodes number, transfer function and the optimal network weights.

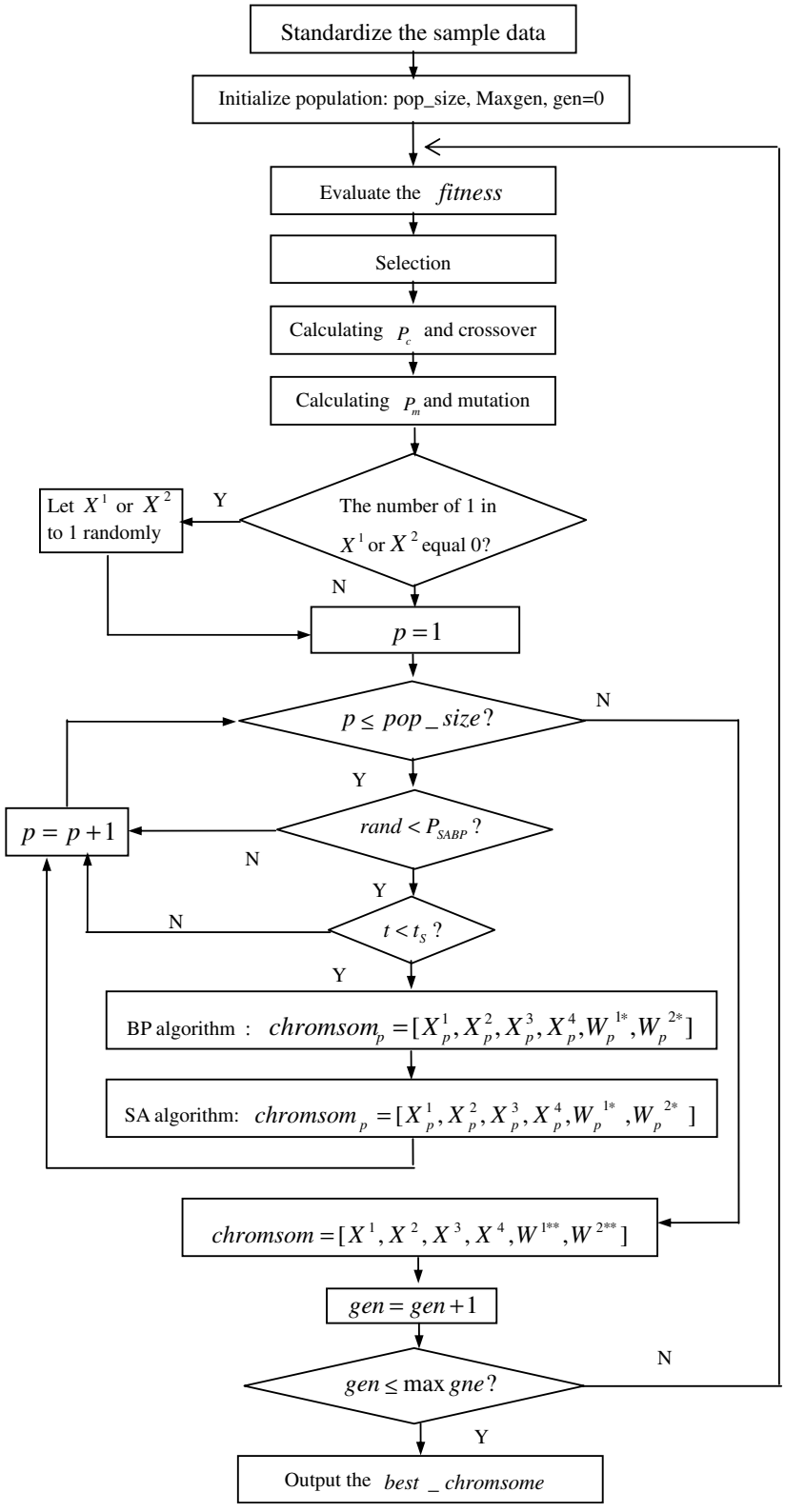

Fig. 4. The flowchart of GA–SA–BP.

Table 1
The samples and raw data

Seismic attribute	Oil well														
	G53	G101	G104	G105	G108	G207	G406	G508	G43	G405	G34	G35	G36	G38	G23
Average peak value	0.07	−0.45	0.33	0.01	0.01	0.53	−0.11	0.17	−0.03	0.11	0.12	0.08	0.03	0.12	0.02
Average trough value	−0.28	−1.24	−1.07	−1.49	−1.59	−0.29	−0.80	−2.66	−0.48	−1.80	−0.89	1.87	−0.98	−1.76	−1.51
Max. amplitude	0.13	0.11	0.33	0.01	0.01	0.97	0.00	0.32	0.00	0.11	0.12	0.15	0.03	0.12	0.02
Mean amplitude	−0.1	−0.60	−0.26	−0.53	−0.60	0.05	−0.40	−0.77	−0.32	−0.63	−0.32	−0.71	−0.45	−0.67	−0.58
Min. amplitude	−0.31	−1.35	−1.07	−1.49	−1.59	−0.83	−1.05	−2.66	−0.89	−1.80	−0.89	−1.87	−1.64	−1.76	−1.51
Sum of negative amplitude	−20.11	−35.33	−24.04	−40.37	−29.81	−8.15	−33.17	−35.03	−24.57	−35.75	−21.12	−37.77	−26.90	−35.41	−38.24
Sum of positive amplitude	3.55	7.01	12.17	2.66	2.85	8.12	2.10	17.53	0.78	6.98	7.41	7.77	5.47	9.19	2.82
Arc length	1.15	1.12	1.05	1.22	1.08	1.06	1.21	1.21	1.08	1.15	1.08	1.15	1.16	1.13	1.10
Dominant frequency	116.21	0.00	36.13	0.00	0.00	77.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Average energy	0.03	0.63	0.31	0.53	0.70	0.21	0.25	1.63	0.20	0.87	0.23	1.05	0.50	0.94	0.63
Energy half-time	0.02	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.01	0.02	0.01	0.01
Instantaneous frequency	91.25	40.39	37.32	37.97	37.54	88.83	45.39	37.10	35.19	39.01	40.15	37.95	42.88	37.98	38.67
Instantaneous phase	0.04	−0.50	0.12	0.68	0.23	−0.62	−0.90	0.20	−0.44	−0.02	0.57	0.11	0.57	0.11	0.19
Average magnitude	1.24	1.84	1.45	2.15	1.48	0.63	1.68	2.29	1.27	2.03	1.24	2.07	1.35	2.03	2.05
Max. magnitude	2.85	2.74	2.42	5.07	3.11	1.58	3.93	4.67	2.45	3.85	2.70	3.63	3.13	3.59	3.73
Ratio of pos to neg sample	46.15	27.78	66.67	17.65	37.50	85.71	10.53	76.92	11.11	50.00	53.33	57.14	26.32	37.50	17.65
Sandstone thickness	2.2	9.20	6.80	8.30	6.80	20.40	2.20	20.00	11.20	5.40	12.40	5.20	24.00	4.00	11.60

The GA–SA–BP NN model is not a simple combination with the above steps, but a mutual integration. Three algorithms will exert their respective merits. Dynamic all parameters of GA–SA–BP start from many initial points simultaneously to search for a group of nodes in the solution space, construct a constantly evolving population serial, and combine the random and directional searching to implement win–win. At the time being, individual go to SA–PB based on certain probability, but not all individuals in the non-population. This could both assure precision of the network, and greatly improve the operation speed of the model. A flowchart of the describing approach is presented in Fig. 4.

4. Application on oil reservoir prediction

In reservoir engineering, the application of reservoir lateral forecasting technique has become a main approach in the exploration and development of oil and gas field. It can obtain believable results on reservoir depth, thickness, and physical properties, utilizing seismic and log data, and tracking the horizontal distribution of reservoir. ANNs approaches commonly provide results that are statistically more significant than "conventional" methods such as multivariate regression, although the nature of the relationships between attributes and reservoir thickness is more difficult to evaluate with ANNs results [27–29]. A simulation study to predict the thickness of oil reservoir based on the seismic information using the dynamic all parameters GA–SA–BP model.

4.1. Network input and sample

The major tool to predict the thickness of oil reservoir is the seismic information. More than 50 seismic attributes which are specific measurements of geometric, kinematics, dynamic, or statistical features derived from seismic data could be gotten. Not all of them have definite corresponding relationship with the thickness of reservoir; obviously, some of the seismic attributes may have no relationship with the thickness of reservoir at all, which results in repetition and redundancies of information. The infinite increase of the number of attributes brings bad effects on reservoir prediction, and the seismic attributes sensitive to targets predicted are not completely identical in different work areas and different reservoirs. For that reason, seismic attributes have to be optimized properly.

In this article, we select an oil zone which belongs to JiangHan basin, center of China and extract 15 seismic attributes in certain window length along the line 167 which have 15 wells on or near. The seismic attribute values will be used as inputs sand the thickness of reservoir will be the target, and values are inputted into GA–SA–BP to optimize the seismic attributes which is the appropriate number of inputs. Table 1 presents the samples and raw data. Randomly select 12 of the 15 wells as the sample training network, and three wells as the testing sample.

4.2. Comparative researches on network performance

According to the model design in Section 2 and the procedures of algorithm in Section 3, 20 random experiments would be done, when the training reaches the max gen, and MSE is 1.2323e−005. During the 20

Table 2
The parameters setting and the performance comparison of 20 randomly experiments with various existing methods

ANN algorithms	Genetic generation	pop_size	t_0	L	Iterations of BP training	Mean of training MSE	Mean of testing MSE
Momentum with adaptive learning rate BP [29][a]	–	–	–	–	2500	1.7830e−004	0.1254
GA–SA [30][a]	10	50	50	20	–	4.1201e−004	0.1167
GA–BP [30][a]	10	50	–	–	100	8.7420e−005	0.1906
SA–BP [31][a]	–	–	50	20	100	8.5731e−005	0.1308
GA–SA–BP with a fixed structure[a]	10	50	50	20	100	7.4458e−005	0.1845
Adaptive dynamic all parameters GA–SA–BP	10	50	50	20	100	3.9709e−005	0.0315

[a] Represents the designed input node number is the largest dimension 15, hidden node number is 8, hidden layer and output layer transfer function randomly selected initially, the topology structure is fixed, namely 15–8–1.

random experiments with the same parameters setting, there will be 18 times to get the best individual corresponding network which is 6–3–1, namely the structure are six input nodes, three hidden nods and one output node. The transfer functions of hidden layer and output layer are tansig and logsig respectively. And do a comparative research of all dynamic parameters GA–SA–BP network model and the Momentum with adaptive learning rate BP model, GA–SA training network model, GA–BP training network model and SA–BP training network model. The setting of parameters and the results of 20 random experiments are listed in the Table 2. We can get the result that dynamic all parameters GA–SA–BP model has better precision and generation.

5. Conclusion

(1) Utilizing multi-code which combine the GA, SA and BP perfectly, and each of them can exert their individual merits and realize win–win. Because the dynamic all parameter GA–SA–BP make use of GA population to do the research and SA probability combination; random conjunction of GA, SA and BP directional search, it is easier to get the global optimization.
(2) Let individual enter SA–BP by certain probability P_{SABP}, this heuristic method could greatly improve the speed of calculation, and also assure the convergence of algorithm.
(3) The relative parameter in the BP network design will be adapted to optimize the topology structure, transfer function based on the sample characters in the learning process and this could greatly reduce the subjectivity and get the simplest network structure with certain precision and also strengthen the generation of network.

Acknowledgement

This research was fully supported by National Natural Science Foundation Grant No. 70573101 of the People's Republic of China.

References

[1] T. Statake, M. Katsumi, N. Nakamura, Neural network approach for minimizing the makespan of the general job-shop, International Journal of Production Economics 33 (1994) 67–74.
[2] J. Bode, Neural networks for cost estimation: simulations and pilot application, International Journal of Production Research 38 (6) (2000) 1231–1254.
[3] I. Sabuncuoglu, B. Gurgun, A neural network model for scheduling problems, European Journal of Operational Research 93 (1996) 288–299.
[4] Q.M. Zhu, A back propagation algorithm to estimate the parameters of non-linear dynamic rational models, Applied Mathematical Modeling 27 (2003) 169–187.
[5] A.G. Olabia, G. Casalino, An ANN and Taguchi algorithms integrated approach to the optimization of CO_2 laser welding, Advances in Engineering Software 37 (2006) 643–648.
[6] Y.L. Loukas, Artificial neural networks in liquid chromatography: Efficient and improved quantitative structure–retention relationship models, Journal of Chromatography A 904 (2000) 119–129.
[7] J.F.C. Khaw, B.S. Lim, L.E.N. Lim, Optimal design of neural networks using the Taguchi method, Neurocomputing 7 (1995) 225–245.
[8] H.R. Maier, G.C. Dandy, The effect of internal parameters and geometry on the performance of back-propagation neural networks: an empirical study, Environmental Modeling and Software 13 (1998) 193–209.
[9] H.R. Maier, G.C. Dandy, Understanding the behavior and optimizing the performance of back-propagation neural networks: an empirical study, Environmental Modeling and Software 13 (1998) 179–191.
[10] P.G. Benardos, G.-C. Vosniakos, Prediction of surface roughness in CNC face milling using neural networks and Taguchi's design of experiments, Robotics and Computer Integrated Manufacturing 18 (2002) 343–354.
[11] J.P. Ross, Taguchi Techniques for Quality Engineering, McGraw- Hill, New York, 1996.
[12] Wanmei Tang, The study of the optimal structure of BP neural network, Systems Engineering – Theory and Practice 10 (2005) 95–100.
[13] S.E. Fahlman, C. Lebiere, The Cascade-Correlation Learning Architecture, Advances in Neural Information Systems, vol. 2, Morgan- Kaufmann, Los Altos, CA, 1990.
[14] S.D. Balkin, J.K. Ord, Automatic neural network modeling for univariate time series, International Journal of Forecasting 16 (2000) 509–515.

[15] M.M. Islam, K. Murase, A new algorithm to design compact two hidden-layer artificial neural networks, Neural Networks 14 (2001) 1265–1278.
[16] X. Jiang, A.H.K.S. Wah, Constructing and training feed-forward neural networks for pattern classification, Pattern Recognition 36 (2003) 853–867.
[17] L. Ma, K. Khorasani, A new strategy for adaptively constructing multilayer feed forward neural networks, Neurocomputing 51 (2003) 361–385.
[18] P.A. Castillo, J.J. Merelo, A. Prieto, V. Rivas, G. Romero, GProp: global optimization of multilayer perceptrons using gas, Neurocomputing 35 (2000) 149–163.
[19] J. Arifovic, R. Gencay, Using genetic algorithms to select architecture of a feed forward artificial neural network, Physica A 289 (2001) 574–594.
[20] N. Harri, H. Teri, et al., Evolving the neural network model for forecasting air pollution time series, Engineering Applications of Artificial Intelligence 17 (2004) 159–167.
[21] Guangzheng Zhang, Deshuang Huang, Prediction of inter-residue contacts map based on genetic algorithm optimized radial basis function neural network and binary input encoding scheme, Journal of Computer-Aided Molecular Design 18 (2004) 797–810.
[22] Shujuan Li, Yan Li, A GA-based NN approach for makespan estimation, Applied Mathematics and Computation 185 (2007) 1003–1014.
[23] S. Kirkpatrick, C.D. Gerlatt Jr., M.P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.
[24] Timo Poranen, A simulated annealing algorithm for determining the thickness of a graph, Information Sciences 172 (2005) 155–172.
[25] Zhang Li, S. Ganesh, An evaluation of back-propagation neural networks for the optimal design of structural systems: Part I. Training procedures, Computer Methods in Applied Mechanics and Engineering 191 (2002) 2873–2886.
[26] M. Srinivas, L.M. Patnaik, Adaptive probabilities of crossover and mutation in genetic algorithms, IEEE Transaction on System, Man and Cybernetic 4 (1994) 656–667.
[27] R.S. Balch et al., Using Artificial Intelligence to Correlate Multiple Seismic Attributes to Reservoir Properties, SPE 56733, 1999.
[28] O. Tapias, Reservoir Engineer and Artificial Intelligence Techniques for Data Analysis, SPE 68743, 2001.
[29] Yibao Li, Xueyong Zhang, et al., Study of improving algorithms based on the BP neural network [J], Journal of Hefei University of Technology 6 (2005) 668–671.
[30] Ling Wang, Intelligent Optimization Algorithms with Applications, Tsinghua Press, Beijing, 2001.
[31] Fujun Hou, Qizong Wu, Forecast on temporal sequence of railway freight transport volume based on BP–SA mixing and optimizing solution, Railway Transport and Economy 10 (2003) 51–53.