Research and Application of Hybrid PSO-BP Neural Network

In fracture acidizing well production prediction

Shuo Zhang, Hongjun Li, Lin Wang, Dezheng Liu, Peng Zou, Enshun Ping, Tianli Ma, Qi Huang

CNPC Bohai Drilling Engineering Company Limited, Tianjin 300450, China

Abstract

In this paper, the limitations of conventional BP algorithm were analyzed, and to enhance its generalization capability of the network, the PSO (particle swarm optimization) algorithm was used to optimize the initial weights of nodes in BP neural network and overcome the over-fitting problem and the local minimum problem of the BP neural network. A new algorithm PSO-BP was studied by giving full play to both of the PSO algorithm's global optimization ability and BP algorithm's local search advantage, the model can overcome the slow convergence and easily getting into the local extremum of basic BP algorithm, and can also improve the learning ability and generalization ability with a higher precision. To demonstrate the capacity of the proposed model, we applied the model to the prediction of the fracture acidizing well production. The mean prediction accuracy of the PSO-BP algorithm is 93.56%, and the mean prediction accuracy of the BP algorithm is 87.73%. The results show that the prediction system based on PSO-BP network is more accurate than traditional BP network, so the algorithm based on PSO-BP network model provides a more accurate, safe and reliable result for the fracture acidizing well production prediction.

Keywords: particle swarm optimization; fracture acidizing well production; prediction; BP neural network; precision

1. INTRODUCTION

The prediction of the fracture acidizing well production is the premise of fracture acidizing, meanwhile it is also extremely significant to optimize each work link during exploratory boring and development process in oilfield. At present, artificial neural network (ANN) has been applied in the prediction of the fracture acidizing well production. The significant research achievements are obtained without any complicated mathematical analyses. Many fracture acidizing well production prediction methods could predicate the fracture acidizing well production but the methods have certain disadvantages (Zhou et al., 2007), such as EL neural network, BP neural network, grey evaluation, multiple discriminating analysis and fuzzy mathematics (Ye et al., 2000), so we should search a new prediction methods for the fracture acidizing well production. Elman neural network can be regarded as a feedfoward network with local memory unit and feedback connections. Its main structure is feedforward connection to memory the output values before a moment, and the connection weights are fixed to cause convergence speed slow. Gray system theory is a new method to study uncertainty problems with poor data and information. And model method possesses advantages of small amounts of data to need and little computation, as a result, it is applied in a wide range among some
certain industries. In grey model, we term the planar clipped between upper bound and lower bound of forecasted future value as grey panel. The planar presents a trumpet, that is, the far the future moment is, the large prediction grey intervals shows, i.e., the practical significance of predicted values is inversed to the grey level.

BP neural network is successfully applied to complex nonlinear problems in fracture acidizing well production prediction by virtue of the strong uniform continuous function approximation of any three layers BP, whilst it bears the powerful ability of self-learning, self-adaptive and nonlinear mapping. Although the BP network in terms of nonlinear system prediction can be widely used, there are still some flaws and lacks, include the following aspects.

a) The learning rate is fixed, so the networks slow convergence requires a longer training time. For some complex problems, BP algorithm training time required may be very long.

b) BP algorithm can converge to a value of weight, but does not guarantee the global minimum of the error surface; this value may be a local minimum.

c) The network hidden layer unit number of the selected layers and there is no theoretical guidance, generally based on experience or determined by trial and error. Therefore, there is often great network redundancy, to a certain extent, also increased the burden of learning.

To overcome these shortcomings, a combination of an intelligent evolutionary algorithm was used to train and optimize the neural networks structure. The intelligent techniques, such as particle swarm optimization (PSO), genetic algorithm (GA) and ant colony algorithm (ACA), have been developed rapidly (Van and Nien, 1992). They have achieved important progresses in parameter optimization, fuzzy simulation and problem solving of acceptableness observation data, etc (Guo and Gong, 1999). In this paper, PSO was used to optimize the initial weights of nodes in BP neural network and overcome the over-fitting problem and the local minima problem of the BP neural network. In order to improve control performance and prediction accuracy, this paper proposed the PSO and BP hybrid algorithm (PSO-BP algorithm) optimization neural network, when it is used in nonlinear model predictive, have advantages of short time-consuming, high accuracy. The test results show that the accuracy of PSO-BP algorithm is significantly higher than that of the conventional BP algorithm method. So the algorithm based on PSO-BP network model provides a more accurate, safe and reliable result for fracture acidizing well production prediction (Che, 2010).

2 Methods Description

2.1 BP Neural Network

BP neural network is a multi-layer feedforward neural networks that is presented in 1986 by Rumelhant and McClelland, which is made of input layer, hidden layer and output layer (Guo et al., 1996). The learning process of the BP neural network is composed by forward propagation and back propagation. In the forward propagation, the input signal is transferred from the input layer to output layer through hidden layer, if the output layer can not get the desired one, then it will turn to back propagation, the error signal will return to the original connection and the weights and thresholds will be adjusted by constantly training until the network error is reduced to the required accuracy. A classical 3-layer BP network model as Fig. 1:
Figure 1. A classical 3-layer BP network model

where \( n \) is the input layer nodes, \( l \) is the hidden layer nodes, \( m \) is the output layer nodes, \( w_{ij} \) called input-hidden layer link weights, \( w_{jk} \) called output-hidden layer link weights.

Then, the output value of hidden layer is:

\[
H_j = f\left(\sum_{i=1}^{n} w_{ij}x_i - \theta_j\right), j = 1,2,...,l
\]  

(1)

Where, \( \theta_j \) is the threshold for the hidden layer nodes, \( f(\cdot) \) is a nonlinear transfer function, generally use the Sigmoid function, that is:

\[
f(x) = \frac{1}{1 + e^{-x}}
\]  

(2)

Thus the actual outputs of network can be obtained:

\[
O_k = \sum_{j=1}^{l} H_j w_{jk} - a_k, k = 1,2,...,m
\]  

(3)

where \( a_k \) is the threshold for the output layer nodes.

According to the desired actual outputs \( O \) and outputs \( Y \), we can get the network error \( e \), the formula is:

\[
e_k = \frac{1}{2} \sum_{k}^{m} (Y_k - O_k)^2, k = 1,2,...,m
\]  

(4)
In the training process, weights and thresholds are adjusted to minimize $e$, which based on the following equation.

The update formula of weights is:

$$w_j = w_j + \eta H_j (1-H_j) x(i) \sum_{k=1}^{m} w_{j_k} e_k$$  \hspace{1cm} (5)

$j=1,2,...,n; \ k=1,2,...,l$;

$$w_{j_k} = w_{j_k} + \eta H_j e_k$$  \hspace{1cm} (6)

$j=1,2,...,l; \ k=1,2,...,m$;

The update formula of thresholds is:

$$\theta_j = \theta_j + \eta H_j (1-H_j) \sum_{k=1}^{m} w_{j_k} e_k, \ j = 1,2,...,l$$  \hspace{1cm} (7)

$$a_k = a_k + e_k, \ k = 1,2,...,m$$  \hspace{1cm} (8)

Where $\eta$ is the learning rate, which is a normal number and always less than 1.

**2.2 PSO-based training algorithm**

PSO algorithm is an evolutionary computation technique based on the swarm intelligence theory, which is originally proposed by James Kennedy and Russell Eberhart. The original intent was to graphically simulate the predatory behavior of bird flock. PSO is a class of derivative free metaheuristic search algorithms inspired by the intelligent co-operation between individual organisms in a group of animals.

It was designed to simulate the co-ordination and movement dynamics in biological swarms as they search for sources of food or avoid adversaries (Eberhart and Shi, 2000). Unlike evolutionary optimization algorithms, such as genetic algorithm (GA) and evolutionary programming (EP), which are based on competition and survival of the fittest, PSO is based on information sharing between individual members of the population. Despite the simplicity of its algorithm in comparison with GA and EP, a comparative study revealed that PSO algorithms perform better than both GA and EP (Boeringer and Werner, 2004). The artificial swarm consists of a population of particles, each of which is described by its position (co-ordinate) and its velocity. The particle co-ordinates are a set of parameters representing a potential solution to the optimization problem at hand. A particle aim sat a better position in the search domain by moving away from its current position. The velocity of this movement is dictated by a combination of actors, including the current momentum of the particle, its past experience, and the experience of the best positioned particle in the swarm. Particles acquire experience by keeping in memory the co-ordinates of the best position they have ever visited in their search history. To find the optimum solution, the model begins by randomly positioning each particle in the swarm.
The performance of each particle is evaluated using a predefined fitness function, which encapsulates the characteristics of the optimization problem, and along with a velocity deciding the direction and distance of its flight. In the search process, the location of particle will be dynamically updated in accordance with its own flying experience and that of neighboring particles, until reach the best position (Che, 2010). That is, we can find the optimal solution through the cooperation among particles in group.

Assume there is a N-dimension search space, $S \subset \mathbb{R}^n$, and the swarm consisting of m particles. Let $X_i=(x_{i1}, x_{i2}, \ldots, x_{in})^T \in S$ be the $i$th particle in the swarm, and the current velocity of this particle is denoted by $V_i=(v_{i1}, v_{i2}, \ldots, v_{in})^T \in S$. Suppose that the current best previous position of the $i$th particle is represented as $P_i=(p_{i1}, p_{i2}, \ldots, p_{in})^T \in S$, called individual extremum, and the current best global position of the whole group is recorded as $P_g=(P_{g1}, P_{g2}, \ldots, P_{gn})^T \in S$, called global extremum. With the above assumptions, the swarm is updated as follows:

Let $f: S \rightarrow \mathbb{R}$ be the target function which will be minimized. At the $k$th iteration, the best position $P_i$ is determined by:

$$P_{i}^{k+1} = \begin{cases} P_i^k, & f(X_i^{k+1}) \geq f(P_i^k) \\ X_i^{k+1}, & f(X_i^{k+1}) \leq f(P_i^k) \end{cases}$$

(9)

And the best position $P_g$ is determined by:

$$P_g = \max \left\{ f(P_i^k) \middle| P_i^k \in S, i = 1, 2, \ldots, m \right\}$$

(10)

Then, the velocity and position will be dynamically adjusted according to the following formula:

$$V_i^{k+1} = w \cdot V_i^k + c_1 \cdot \text{rand} \cdot (P_i^k - X_i^k) + c_2 \cdot \text{rand} \cdot (P_g^k - X_i^k)$$

(11)

$$X_i^{k+1} = X_i^k + V_i^{k+1}$$

(12)

where, $c_1$ and $c_2$ are constants and called as acceleration factor; rand is random numbers changing in the range from 0 to 1; $w$ is the inertia weight, which is responsible for balancing between local search and global search, and improving the convergence performance of the algorithm. Usually, $w$ is usually calculated by the following formula:

$$w = w_{\text{max}} - \frac{w_{\text{max}} - w_{\text{min}}}{\text{iter}_{\text{max}}} \cdot \text{iter}^k$$

(13)

where $w_{\text{max}}$ and $w_{\text{min}}$ are maximum weighted coefficient and minimum weighted coefficient; $\text{iter}_{\text{max}}$ is the maximum number of iteration, and $\text{iter}^k$ is the current number of iteration.

### 2.3 PSO–BP hybrid algorithms

In recognition of the performance of PSO in global searches and BP algorithms in local searches, hybrid PSO–BP algorithms have been developed with the aim of achieving an efficient global search without compromising the local search.
Being a continuous optimization technique, PSO is ideally suitable for network training. It was reported in the earliest paper on PSO-based neural net training that a particles warm optimiser could train NN weights as effectively as the usual error back-propagation method (Clerc and Kennedy, 2002). Later, Liu compared the performance of PSO and BP using simulated data (a quadratic curve) (Liu, 2003). Their result indicated that PSO converges faster than back-propagation. Despite its greater ability to traverse the search space in pursuit of a global optimum, PSO, like other metaheuristic search algorithms, are inferior in local searches to gradient-based algorithms. Jumping over the best solution and converging to some point away from the global optimum is not uncommon with metaheuristic search methods, due to their lack of a focus on local minima.

Zhang et al. developed a hybrid algorithm in which an ANN is constructed using PSO until no improvement in accuracy is obtained (Zhang et al., 2000). Back-propagation is then used to complete the training. From their results, a combination of BP and PSO improved the network performance over that of either BP or PSO, both in terms of the convergence speed and the error level. Huang et al. used a similar BP-PSO structure to perform a nonlinear approximation with remarkably good results (Huang et al., 2009). The risk associated with the hybrid algorithms in the literature, notwithstanding the good results reported, is that PSO, which performs the first phase of optimization, may converge to some point far away from a global optimum due to the stochastic nature of the tuning parameters. This could make it difficult for the BP algorithm to reach the best solution when it takes over to conduct the second phase of the search. Developing a hybrid algorithm which optimizes the use of both the BP and PSO methods of training algorithm, therefore significantly improve the quality of the search, both locally and globally.

In this paper, the proposed hybrid optimization algorithm is based on the PSO technique and BP neural network, which are both used successively as the training progresses. The idea is to get the best out of these two powerful algorithms by developing an algorithm which integrates more efficiently the PSO and BP algorithms. The algorithm involves initially training the network using PSO for a certain number of iterations, then training some selected particles from among the swarm population using the BP algorithm for a few steps. The results of the local search by the BP algorithm are then used to update the positions of the relevant particles and PSO takes over again. This cycle is repeated until a sufficiently accurate output is obtained.

In the searching process of BP neural network which is trained by PSO algorithm, the position of each particle in the swarm will be represented as the weights and thresholds of BP network. The network error \(e\) between actual output and desired output is defined as the fitness function in the training. Then, the smaller the training error is, the better the particles perform in the search process, and the optimal solution is the particle that has the smallest network error \(e\).

The algorithm flow is as follows:

Step 1: Establish BP neural network according to the practical problems, and determine the number of nodes in each layer (include input layer, output layer and hidden layer), initial weights and thresholds, given the value of learning rate \(\eta\) and the desired outputs \(Y\).

Step 2: Initialize swarm, including swarm size, randomly initialize the local optimal value \(P_i\) and the global optimal value \(P_g\), given the value of \(w_{max}\), \(w_{min}\), \(c_1\), \(c_2\) and \(rand\), the maximum number of iteration in PSO.

Step 3: Select training sample, calculate the fitness value of each particle using network error.
function which is showed as formula (4).

Step 4: Compare and then update the velocity and position in accordance with the formulas (11) and (12).

Step 5: Determine it whether meets the conditions to terminate the algorithm. If the fitness of each particle meets the error requirement or maximum number of iteration, then run to step 6, else to step 3.

Step 6: Output the optimized weights, the optimal solution can be obtained.

3 Applications

3.1 Influencing Factors

According to the viewpoint of materialist dialectics, the external performance of material is its internal properties embodiment, and the properties of matter also have certain relation. If we regard the fracture acidizing well production as external embodiment of reservoir internal properties, well then, the fracture acidizing well production is certainly relative to properties of minerals, pore space and fluid (Zhan et al., 2015). As a result, we should firstly study the relation between internal properties and sensitivity of reservoirs, in order to find out the main internal factors influencing the fracture acidizing well production (Qiu et al., 2005). There are several factors influencing the fracture acidizing well production prediction, and the mutual action to each other is complex (Zhou et al., 2007). After analyzing reservoir characteristics and experimental data from the oilfield, it is considered that the fracture acidizing well production is closely related to many factors, such as porosity, permeability, clay minerals content and so on, the influencing factors of sensitivity damage are found out. Based on the analysis of the principles of the fracture acidizing well production, the influence factors of the fracture acidizing well production were obtained. The main influence factors of the fracture acidizing well production are shown in Table 1.

<table>
<thead>
<tr>
<th>Influencing factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>fracture acidizing well production Porosity, Permeability, Clay minerals content, Chlorite content, Cementation type, Chlorite/smectite mixed layer content, Quartz content, Particle sorting characteristics, Carbonate content, Shale content</td>
</tr>
</tbody>
</table>

In this paper, we select the influencing factors of the fracture acidizing well production prediction as input variables, the corresponding sensitivity index as the output variables.

3.2 Pre-processing of Data

The raw data of the factors contain qualitative data (such as cement type and particle sorting characteristics) and quantitative data. The quantitative data of different influence factors may be quite different. So it is necessary to process the raw data to ensure the values of all the influence factors be limited in a small range between 0 and 1. The processing of cementation type applies the expert knowledge based on the theory of deposition (Sun et al., 1999). The assignment of different cementation type is shown in Table 2.
Table 2 Assignment of different cementation type

<table>
<thead>
<tr>
<th>NO.</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Film cementation</td>
<td>0.1892</td>
</tr>
<tr>
<td>2</td>
<td>Contact cementation</td>
<td>0.2703</td>
</tr>
<tr>
<td>3</td>
<td>Pore-contact cementation</td>
<td>0.3514</td>
</tr>
<tr>
<td>4</td>
<td>Basal-contact cementation</td>
<td>0.4054</td>
</tr>
<tr>
<td>5</td>
<td>Pore-pack cementation</td>
<td>0.4865</td>
</tr>
<tr>
<td>6</td>
<td>Pore cementation</td>
<td>0.5405</td>
</tr>
<tr>
<td>7</td>
<td>Basal-pore type cementation</td>
<td>0.6216</td>
</tr>
<tr>
<td>8</td>
<td>Pack cementation</td>
<td>0.7027</td>
</tr>
<tr>
<td>9</td>
<td>Basal cementation</td>
<td>0.8378</td>
</tr>
</tbody>
</table>

The processing of particle sorting characteristics applies the Fokker Ward sorting standard deviation, which can be expressed as formula (14):

\[
\Gamma_i = \begin{cases} 
0.0690; & \delta_i < 0.35 \\
\delta_i / 4.35; & 0.35 \leq \delta_i \leq 4.0 \\
0.9432; & \delta_i > 4.0 
\end{cases} \tag{14}
\]

where \(\Gamma_i\) is the normalized value of particles sorting coefficient after processing, and \(\sigma_i\) is the Fokker Ward sorting standard deviation of particle sorting characteristics.

To eliminate the deviation derived from different values, it is necessary to normalize the data using the arithmetic mean method. After normalizing the same factor, it has the minimum boundary 0 and the maximal boundary 1. The mean value can be calculated by the following formula (15):

\[
T = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{15}
\]

\(T\) indicates a group of data which has been normalized. \(x\) is defined as an initial group of data among the collected data groups. \(x_{\min}\) and \(x_{\max}\) express the minimum and maximum data group among the collected data groups.

### 3.3 The Number of the Neurons in Hidden-layer

In this paper, BP network has three layers including one hidden-layer. The number of the neurons in hidden-layer affects the performance of neural networks. We can determine the number of neurons in hidden-layer by comparing the precision between different schemes. First, experiential formula (16) is used to calculate the hidden layer neuron number:

\[
\begin{align*}
I & = \sqrt{m + n + a} \\
\end{align*} \tag{16}
\]

Where, \(I\) is the hidden layer neuron number, \(n\) is the input factors number, \(m\) is the output factors number, \(a \in [0, 10]\) is an integer constant, formula (16) is used to calculate the approximate range, and then the predicted accuracy of different number of neurons in the
range is compared. The number of neurons which has the most accurate prediction is the suitable number. In this paper, the suitable number of neurons is shown in table 3.

### Table 3 Number of the neuron nodes in different layers

<table>
<thead>
<tr>
<th>Input layer number</th>
<th>Hidden layer number</th>
<th>Output layer number</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>21</td>
<td>1</td>
</tr>
</tbody>
</table>

The selected parameters for the PSO algorithm are set as follows: number of particles in swarm $m=20$; maximum number of iterations $\text{itermax}=100$; learning factor $c_1=2.05$, $c_2=2.05$; maximum inertia weight $w_{\text{max}}=0.9$, minimum inertia weight $w_{\text{min}}=0.4$; training error $e=10^{-3}$.

### 4 Results and discussion

In this study, the fracture acidizing well production prediction data was collected from our own experiments and processed to train and test neural network. In order to estimate the accuracy of PSO-BP neural network in prediction of the fracture acidizing well production, the PSO-BP neural network and the BP neural network were trained and tested respectively by the same input data. The test steps are taken as follows: One sample was taken as the test sample and the other samples as training samples; the sensitivity index of each training sample was taken as expected output while the corresponding reservoir property data as the input layer neurons to train the neural network, finally, the network was tested by the sample that has not been used to train the network. The steps above were repeated until the network was tested by all the samples. We make analysis and forecast for the fracture acidizing well production of known samples, and compare the measured results with predicted ones, and the fracture acidizing well production predicted results are shown in Fig. 2.

![Figure 1. Prediction values of PSO-BP algorithm and BP algorithm.](image-url)

Fig. 2 shows the actual sensitivity indexes (the normalization values of the fracture acidizing well production) those were measured in the laboratory in comparison with the prediction results. By comparison with the BP neural network, it is obvious that the PSO-BP neural network has better approximation ability and higher prediction accuracy.
Fig. 3 shows the the fracture acidizing well production prediction indexes accuracy of the PSO-BP algorithm and the BP algorithm. From Fig.3, we can know that the predicted accuracy of PSO-BP algorithm is much higher than that of BP algorithm. The mean prediction accuracy of the PSO-BP algorithm is 93.56%, and the mean prediction accuracy of the BP algorithm is 87.73%. As a result, it comes to a conclusion that the predicted accuracy of PSO-BP network is obviously better than that of BP network. Therefore, PSO-BP neural network is very attractive for a wide application in forecasting the the fracture acidizing well production.

5. CONCLUSION

In this paper, the particle swarm optimization (PSO) algorithm is introduced to optimize the weight and threshold of the neural network, then, a model of the the fracture acidizing well production forecasting based on PSO algorithm and BP neural network integrated is proposed to improve the efficiency of BP neural network in global search and fine tuning of the solutions, and also, very competitive results have been verified by PSO-BP network in comparison with BP algorithms. The mean prediction accuracy of the PSO-BP algorithm is 93.56%, and the mean prediction accuracy of the BP algorithm is 87.73%. The experimental results confirmed that the PSO-BP method is better than the traditional BP neural network. Therefore, the model based on PSO-BP neural network is effective and feasible for the fracture acidizing well production prediction which can be applied in petroleum industry and other fields.

6. REFERENCES