A memetic GSA with niching selection for training fuzzy wavelet neural network

Hojjat Allah Bazoobandi, Mahdi Eftekhari
Department of Computer Engineering
Shahid Bahonar University of Kerman, Kerman, Iran.
hojjat.bazoobandi@gmail.com, m.eftekhari@mail.uk.ac.ir

Name of the Presenter: Hojjat Allah Bazoobandi

Abstract
This paper proposes an effective memetic Gravitational Search Algorithm (GSA) that utilizes Solis and Wets' (SW) algorithm as local search. GSA has good exploration ability and SW helps to improve the exploitation ability of the memetic algorithm. Furthermore, a selection strategy is proposed to select suitable individuals for local refinement that is based on subtractive clustering. Proposed memetic algorithm is employed for tuning fuzzy wavelet neural network parameters. We evaluate the performance of the proposed memetic algorithm on two system identification problems. Computational results confirm the performance of the proposed memetic algorithm and selection strategy on training fuzzy wavelet neural network parameters.

Key words: Fuzzy wavelet neural network, Memetic algorithm, GSA, Subtractive clustering.

1. Introduction
Fuzzy Wavelet Neural Network (FWNN) is a powerful tool, which is used for many types of engineering problems such as function approximation, system identification, etc. FWNNs are similar to the common neural networks but fuzzy logic is used in the input layer, and wavelet functions are employed in the hidden layer to improve the performance of common neural networks. The previous researches about FWNN can be divided into two fundamental categories:

- Researches, which tried to modify the structure of FWNN.
- Researches, which introduced new training methods for tuning parameters of FWNN.

For the first category, most of the works try to change the wavelet function of FWNN. For examples, Cao et al. (2010) proposed a WNN that composite functions are applied at the hidden nodes. The resultant composite activation functions are more compact than the ones that use single wavelet functions alone. Cordova and Yu (2012) also applied the Haar wavelet transform as activation functions. Zainuddin and Pauline (2011) compared the approximation ability of WNN when the wavelet functions were changed. They concluded that the Gaussian wavelet is the best one in FWNN modeling. Some other works also introduced about structure of FWNN. Yilmaz and Oysal (2010) proposed three FWNN model. Two of them use summation and multiplication of dilated and translated versions of single-dimensional wavelet basis functions, respectively. In the other one, THEN part of fuzzy rules consists of radial functions of wavelets. Lin et al. (2005) introduced recurrent wavelet based neuro fuzzy
networks for dynamic system identification. The main idea behind this network is adding feedback connections from memory units in the rule layer of the proposed network. Abiyev and Kaynak (2008) proposed a structure which is based on a set of fuzzy rules.

The second and the most important issue about FWNN is the training method. Many researches have been done to address this challenge. Derivative based methods, especially BackPropagation (BP) method, are more employed in training FWNN because of their training speed. For example, Lee and Teng (2000) used BP for training FWNN. Zekri et al. (2008) used Extended Kalman Filter (EKF) for tuning FWNN parameters. Davanipoor et al. (2012) introduced a hybrid scheme. They used Orthogonal Least Square (OLS) for tuning linear parameters of FWNN and BP for nonlinear ones. Although training based on derivative of the objective function have high speed but the solution may trap in local optima. Tzeng (2010) proposed a genetic algorithm for the aim of training FWNN. Wei et al. (2010) trained FWNN with PSO for spatio-temporal system identification problems. Cao et al. (2011) combined Differential Evolution (DE) with Extreme Learning Machine (ELM) to train FWNN.

To this end, we propose a memetic algorithm for training FWNN. Proposed memetic algorithm combines GSA with a classical local search which is named Solis and Wets. Subtractive clustering separates the niches of population and then the local search method was run for appropriate solutions in different niches. The rest of the paper is organized as follows: Section 2 reviews the structure of FWNN. The proposed method represented in Section 3. The performance of proposed memetic algorithm is evaluated in Section 4. Finally, concluding remarks are given in Section 5.

2. FWNN

Abiyev and Kaynak (2008) proposed a structure for FWNN. We used this structure in our study, too. An FWNN in this scheme can be described with a number of fuzzy IF-THEN rules. The form of $i^{th}$ rule presented in the following:

$$ R^i : \text{If } x_1 \text{ is } A^i_1, \text{ and } x_2 \text{ is } A^i_2, \ldots, \text{ and } x_q \text{ is } A^i_q \text{ Then } y_i = w_i \sum_{j=1}^q \psi_{ij}(x_j). $$

(1)

where $x_j (j=1..n)$ and $y_i (i=1..c)$ refer to the input and output signals, respectively. The weight of $i^{th}$ fuzzy rule can be determined as $w_i$. $A^i_j$ shows membership function for $j^{th}$ input of $i^{th}$ rule. $\psi_{ij}$ demonstrate a family of wavelet functions. This family of wavelet functions can be calculate as follows:

$$ \psi_{ij}(x_j) = \psi \left( \frac{x_j - a_{ij}}{b_{ij}} \right), b_{ij} \neq 0. $$

(2)

where $\psi$ represent the mother wavelet function. Mother wavelet is a basis function that other functions in the family could be obtained by translation $(a)$ and/or dilation $(b)$ of it by Eq. 2.

$$ \psi(x) = \frac{1}{\sqrt{|b|}} (1 - 2x^2) exp\left( -\frac{x^2}{2} \right), $$

(3)

Here we used Mexican Hat wavelet function as mother wavelet which can be formulated as follows:

$$ A^i_j(x_j) = exp \left[ -\left( \frac{x_j - c^i_j}{\sigma^i_j} \right)^2 \right]. $$

(4)

where $c^i_j$ and $\sigma^i_j$ represent the center and half-width of membership function. The output of the FWNN model for each input sample can be calculated as follows:
\[ u = \frac{\sum_{i=1}^{c} \mu_i y_i}{\sum_{i=1}^{c} \mu_i} \]  

where \( y_i \) is the output of \( i^{th} \) rule and \( \mu_i \) is the firing strength of \( i^{th} \) fuzzy rule which can be calculated in the following form:

\[ \mu_i(x) = \prod_{j=1}^{n} A^j_i(x_j), \]

### 3. Proposed Memetic Algorithm

Memetic algorithms are population-based meta-heuristic composed of an evolutionary framework and a set of local search algorithms. Neri and Cotta (2012) reviewed the memetic algorithms. They indicate some important aspects of implementation and the coordination of memes. According to these matters, an operational memetic algorithm proposed in this paper. Proposed memetic algorithm employs GSA as an evolutionary framework. Solis and Wets algorithms also used as local search.

#### 1 GSA

GSA is a new evolutionary algorithm which proposed by Rashedi et al. (2009). In GSA, searcher agents are a collection of masses which interact with each other based on the Newtonian gravity and the laws of motion. Each agent can be described with its position and velocity. The position of \( i^{th} \) agent defined as follows:

\[ X_i = (x^1_i, x^2_i, ..., x^{d_i}_i), \quad i = 1, 2, ..., N \]  

where \( x^d_i \) is the position in \( d^{th} \) dimension of the solution space. At a specific time \( t \), the position and velocity of a mass update with the following equations:

\[ V^d_i(t+1) = rand_i \times V^d_i(t) + a^d_i(t), \]

\[ X^d_i(t+1) = X^d_i(t) + V^d_i(t+1) \]

where \( rand_i \) is a random in the interval [0, 1]. The acceleration of the agent \( i \) at time \( t \), and in \( d^{th} \) dimension is given as follows:

\[ a^d_i(t) = \frac{F^d_i(t)}{M_i(t)} \]

where \( M \) and \( F \) shows the mass and randomly weighted sum of the forces extracted from other masses, respectively. The mass of each agent update by the following equation:

\[ m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \]

\[ M_i(t) = \sum_{i=1}^{N} m_i(t) \]

where \( fit_i \) represent the fitness value of \( i^{th} \) agent. The best agent (here the agent with minimum fitness) shows by \( best \) while \( worst \) represent the fitness value of the worst agent. Forces in Eq. 10 can be calculated as follows:

\[ F^d_i(t) = \sum_{j=1,j \neq i}^{N} rand_i F^d_{ij}(t) \]

\[ F^d_{ij}(t) = G \frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \varepsilon} (X^d_j(t) - X^d_i(t)) \]

where \( rand \) is a random number between [0, 1]. \( R_{ij} \) describe the Euclidean distance between two agents \( i \) and \( j \). \( \varepsilon \) is a small constant. \( G \) shows gravitational constant.

#### 2 SW Local Search

Solis and Wets (1981) proposed a randomized hill climber with an adaptive step size. The scheme of SW is presented in Algorithm 1. In SW pseudo-code, \( Sol \) refers to current solution,
$N$ is dimension of solution, and $bias$ is a momentum term that put the search to correct direction.

```
Function SolisWets (Sol, N, bias, rho)
While numEval<maxEval
    For i←1 to N
        dif(i)←RandomGaussian(0, rho)
    EndFor
    Sol’←Sol+bias+dif
    If Sol’ is better than Sol
        Sol←Sol’
        bias←0.2*bias+0.4*(dif+bias);
        numFailed←0
    Else
        Sol’’←Sol-bias-dif
        If Sol’’ is better than Sol
            Sol←Sol’’
            bias←bias-0.4*(dif+bias);
            numFailed←0
        Else
            numFailed←numFailed+1
        EndIf
    EndIf
    If numFailed>3
        rho←rho/2;
        numFailed←0;
    EndIf
EndWhile
EndFunction
```

Algorithm 1: Pseudo-code of Solis and Wets local search.

3 Hybridization methodology
The hybridization scheme in memetic algorithms is very important. Sudholt (2009) analyses the impact of local search frequency on the performance of hybrid algorithm and prove it theoretically. Duan (2013) review eighteen strategy for applying local search. He concluded that a good strategy for applying local search makes the hybrid method much better. Executing the local search over all solutions in memetic algorithms is very time consuming. One of interesting method is finding the niches in population and then local search executed on only the delegate of niches. In this paper we propose a method for applying local search which is based on niches of population. Previous works used clustering methods, like K-Means and Fuzzy-C-Means clustering, such as Martínez et al. (2006). It should be pointed out that in these methods the number of clusters must be predefined. For this purpose, we used subtractive clustering to find niches in solutions. Subtractive clustering is a fast method among clustering methods. Interested reader can find more information about subtractive
clustering in Romera et al. (2007). A radius parameter used in subtractive clustering to separate clusters from each other. We propose the following measure to calculate the radius:

$$radius = \frac{\text{var}(fit)}{2} + \left[1 - \left(\frac{\text{CurrrIter}}{\text{MaxIter}}\right)\right]$$

(16)

where \( \text{var}(fit) \) represents the variance of fitness values. Current and maximum number of iterations demonstrated by \( \text{CurrrIter} \) and \( \text{MaxIter} \), respectively. After applying subtractive clustering on current population and finding niches, a solution is selected from each of niches by known Roulette Wheel mechanism and undergoes to local refinement.

4. Results and Analysis

In this section, we evaluate the performance of the proposed memetic algorithm that is combined with the selection strategy. Root Mean Square Error (RMSE) is used as performance criterion for the aim of straightforward comparison of the proposed method with the others in the literature.

$$\text{RMSE} = \sqrt{\frac{1}{L} \sum_{i=1}^{L} (y(i) - y_d)^2}$$

(17)

100 searching agents used for GSA in simulations.

Example1: As an example, identification of below nonlinear plant using FWNN is considered. The plant is described as:

$$y(k) = 0.72y(k - 1) + 0.025y(k - 2)u(k - 2) + 0.01u^2(k - 3) + 0.02u(k - 4),$$

(18)

where \( u \) is the input signal of the plant. As can be seen, the current output of plant depends on input and output signals. For identification of the plant, the following input signal is used:

$$u(k) = \begin{cases} 
\sin\left(\frac{\pi k}{25}\right), & k < 25 \\
-1, & 250 \leq k < 750 \\
0.3 \sin\left(\frac{\pi k}{25}\right) + 0.1 \sin\left(\frac{\pi k}{32}\right), & 500 \leq k < 750 \\
+0.6 \sin\left(\frac{\pi k}{10}\right), & 750 \leq k < 1000.
\end{cases}$$

(19)

We used 1000 training sample in our simulation. Two fuzzy rules are used in the FWNN structure. The training is continued for 100 iteration of GSA. Fig. 1 shows the evolution of RMSE values over 100 epochs. Fig. 2 compares the actual output of the plant with that of the FWNN identifier. In Table 1, the RMSE values are given for our proposed method and other approaches in the literature. We compare our method with Yilmaz and Oysal (2010) that proposed three FWNN model (FWNN-S, FWNN-R, FWNN-M), and Abiyev and Kaynak (2008) methods (1-FWNN, 2-FWNN).

As can be seen from Table 1, the RMSE value in our proposed method is less than other methods despite considerably smaller number of rules.

Example2: In this example, we evaluate the effectiveness of our memetic algorithm by considering a second order nonlinear plant. The process is described by the following equation:

$$y(k) = f(y(k - 1), y(k - 2), y(k - 3), u(k), u(k - 1))$$

(20)

where function \( f(.) \) is defined as follows:
\[ f(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_3^2 + x_2^2} \] (21)

**Table 1.** Comparison of different models for identification of first dynamic plant.

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of rules</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>FWNN-S</td>
<td>32</td>
<td>0.0097</td>
</tr>
<tr>
<td>FWNN-R</td>
<td>32</td>
<td>0.0096</td>
</tr>
<tr>
<td>FWNN-M</td>
<td>32</td>
<td>0.0096</td>
</tr>
<tr>
<td>1-FWNN</td>
<td>3</td>
<td>0.0197</td>
</tr>
<tr>
<td>2-FWNN</td>
<td>5</td>
<td>0.0187</td>
</tr>
<tr>
<td>Proposed method</td>
<td>2</td>
<td><strong>0.0027</strong></td>
</tr>
</tbody>
</table>

As previous example, 1000 training sample are constructed by the Eq. 19. Two fuzzy rules also are used to describe our FWNN model. Fig. 3 shows the RMSE values obtained during training. Fig. 4 compares the actual output and the output which is obtained by FWNN. Proposed method is compared with Bodyanskiy and Vynokurova (2013) method which proposed a hybrid adaptive method for training FWNN(HA-FWNN), Yilmaz and Oysal (2010) that proposed three FWNN model (FWNN-S, FWNN-R, FWNN-M), Abiyev and Kaynak (2008) methods (1-FWNN, 2-FWNN), and a feed-forward neuro fuzzy system. Table 2 compares the RMSE values with the other methods reported in the literature. It is clear from Table 2 that the performance of our proposed method is much better in training FWNN.

**5. Conclusions**

To enhance the modeling accuracy of FWNN, we propose a memetic algorithm by combining a GSA with “Solis and Wets” (SW) local search algorithm in this paper. GSA makes the proposed memetic algorithm more able for exploration and the SW makes it more powerful in exploitation. For hybridizing these two methods, we propose to use subtractive clustering to find niches and then a suitable solution is selected in each niche to undergo local refinement. Subtractive clustering with proposed radius parameter makes the selection mechanism
adaptive because it is not needed to determine the number of clusters. Experimental results justify the superior performance of our proposed method in terms of generating accurate and compact (i.e. Few number of rules) FWNN models in comparison with previous works.

![Graph](image1.png)  ![Graph](image2.png)

Fig 3: RMSE values during training of example 2.  
Fig 4: Comparison between the actual and predicted output for example 2.

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of rules</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HA-FWNN</td>
<td>-</td>
<td>0.0183</td>
</tr>
<tr>
<td>FWNN-S</td>
<td>32</td>
<td>0.0208</td>
</tr>
<tr>
<td>FWNN-R</td>
<td>32</td>
<td>0.0152</td>
</tr>
<tr>
<td>FWNN-M</td>
<td>32</td>
<td>0.0192</td>
</tr>
<tr>
<td>FWNN</td>
<td>3</td>
<td>0.0291</td>
</tr>
<tr>
<td>FWNN</td>
<td>5</td>
<td>0.0282</td>
</tr>
<tr>
<td>Feed-forward neural fuzzy system</td>
<td>-</td>
<td>0.0203</td>
</tr>
<tr>
<td>Proposed method</td>
<td>2</td>
<td><strong>0.0128</strong></td>
</tr>
</tbody>
</table>

Table 2. Comparison of different models for identification of second dynamic plant.

References


