Teaching Learning Based Optimization Neural Networks for Waveguide Filter Modeling

Chahrزاد Erradir, Mohamed Lahdi Riabi, Halima Ammari, Emir Bouarroudj
Department of Electronics, University Constantine 1
Laboratory of Electromagnetic and Telecommunication
Constantine, Algeria
cerredir@yahoo.fr

Abstract—Artificial neural networks (ANN) provide fast and accurate models for the modeling, simulation, and optimization of microwave components. In this paper, an optimization method, called teaching-learning-based optimization (TLBO) is proposed for training artificial neural networks (ANN). Then, the trained networks are applied to modeling waveguide filter structures. The results obtained using teaching-learning based optimization neural networks (TLBO-NN) algorithms are validated by comparing them with those obtained using particle swarm optimization neural networks (PSO-NN).

Keywords—neural networks; modeling; waveguide filter; Teaching Learning Based Optimization

I. INTRODUCTION

Recently, artificial neural network (ANN) has been proven to be a fast and effective means of modeling complex electromagnetic devices. It has been recognized as a powerful tool for predicting device behavior for which no mathematical model is available or the device has not been analyzed properly yet. ANN can be trained to capture arbitrary input-output relationship to any degree of accuracy. Once a model is developed it can be used over and over again. The trained model delivers the output parameters very quickly. For these attractive qualities, ANN has been applied to different areas of engineering’s [1 – 3].

Training of neural networks is a complex task of great importance in the learning, where it depends on adaptation of free network parameters, that is, on the proper selection of the neural weight values. Specialized learning algorithms are used for adaptation of these weight values. Among those algorithms, the most popular algorithm is a back-propagation method (BP) [4] based on a gradient descending. Lately, many populations based algorithms were proposed for training a neural network such as Particle Swarm Optimization (PSO) [5], Genetic Algorithms [6] and other optimization algorithms [7].

In this paper, we propose a waveguide Filters (Pseudo-Elliptic waveguide filter and H-plane waveguide filters considering rounded corners) modeling using a multilayer neural network trained by a recently proposed algorithm called Teaching-Learning Based Optimization (TLBO) [8–10]. This algorithm is based on the effect of the influence of a teacher on the output of learners in a class. TLBO is a population based algorithm which requires only common parameters (population size and number of generations) and does not require any other specific parameter. To validate the training of neural networks using the proposed algorithm, its results for waveguide microwave structures modeling are compared with the results obtained using other population based algorithm which are widely used in training NN namely Particle Swarm Optimization (PSO).

II. METHODS

A. Particle Swarm Optimization

Particle swarm optimization (PSO) is a stochastic method of optimization based on the reproduction of a social behavior. It was invented by Russell Eberhart and James Kennedy [11] in 1995. They tried to simulate the ability of animal societies that don’t have any leader in their group or swarm (bird flocking and fish schooling) to move synchronously and their ability to change direction suddenly while remaining in an optimal formation (food source). PSO consists of a swarm of particles, where particle represent a potential solution. The particles of the swarm fly through hyperspace and have two essential reasoning capabilities: their memory of their own best position local best (LB) and knowledge of the global or their neighborhood’s best global best (GB). The position and velocity of a particle at each iteration (the passage of time at time tj) are updated using the following Equations

\[ X_i(t+1) = X_i(t) + V_i(t+1) \]  \hspace{1cm} (1)

\[ V_i(t+1) = w \times V_i(t) + c_1 r_1 [LB_i(t) − X_i(t)] + c_2 r_2 [GB(t) − X_i(t)] \] \hspace{1cm} (2)

where \( V_i \) and \( X_i \) are the velocity and the position for particle \( i \) at time \( t \). \( w \) is the inertia weight, at each iteration update with the following Equation [12]

\[ w(t) = w_{max} - (w_{max} - w_{min}) \times \frac{t}{\maxit} \] \hspace{1cm} (3)

The parameters \( w_{max} \), \( w_{min} \), \( \maxit \). \( c_1 \) and \( c_2 \) are constant coefficients determined by the user. \( r_1 \) and \( r_2 \) are random numbers between 0 and 1.
B. Teaching Learning Based Optimization

In 2011, Rao et al. [8―10] proposed an algorithm, called Teaching-Learning-Based Optimization (TLBO), based on the traditional Teaching Learning phenomenon of a classroom. TLBO is a population based algorithm, where a group of students (i.e. learner) is considered as population and the different subjects offered to the learners are analogous with the different design variables of the optimization problem. The results of the learner are analogous to the fitness value of the optimization problem. The best solution in the entire population is considered as the teacher. Teacher and learners are the two vital components of the algorithm, so there are two modes of learning; through the teacher (known as the teacher phase) and interacting with other learners (known as the learner phase).

1) Teacher Phase: In this part, learners take their knowledge directly through the teacher, where a teacher tries to increase the mean result value of the classroom to another value, which is better than, depending on his or her capability. This follows a random process depending on many factors. In this work, the value of solution is represented as \( X_{j,k,i} \), where \( j \) means the \( j \)th design variable (i.e. subject taken by the learners), \( k = 1, 2, ..., m \); \( k \) represents the \( k \)th population member (i.e. learner), \( k = 1, 2, ..., N \); and \( i \) represents the \( i \)th iteration, \( i = 1, 2, ..., \text{maxIter} \), where maxIter is the number of maximum generations (iterations). The existing solution is updated according to the following expression

\[
X_{j,k,i} = X_{j,k,i} + DM_{j,k,i} \tag{4}
\]

DM \(_{j,k,i}\) is the difference between the existing mean and the new mean of each subject is given by

\[
DM_{j,k,i} = r \times (X_{j,\text{best},i} - TF \times M_{j,i}) \tag{5}
\]

\( M_{j,i} \) is the mean result of the learners in a particular subject \( j \), \( X_{j,\text{best},i} \) the new mean and is the result of the best learner (i.e. teacher) in subject \( j \). \( r \) is the random number in the range \([0, 1]\). \( TF \) The teaching factor is generated randomly during the algorithm in the range of \([1, 2]\), in which 1 corresponds to no increase in the knowledge level and 2 corresponds to complete transfer of knowledge. The in-between values indicates the amount of transfer level of knowledge. The value of \( TF \) is not given as an input to the algorithm its value is randomly decided by the algorithm

\[
TF = \text{round} \{1 + \text{rand}(0, 1) \times (2-1)\} \tag{6}
\]

2) Learner Phase: In this part, learners increase their knowledge by interaction among themselves. A learner interacts randomly with other learners for enhancing his or her knowledge. A learner learns new things if the other learner has more knowledge than him or her. At any iteration \( i \), each learner is compared with the other learners randomly. For comparison, randomly select two learners \( P \) and \( Q \) such that \( X_{j,P,i} \neq X_{j,Q,i} \) (where \( X_{j,P,i} \) and \( X_{j,Q,i} \) are the updated values at the end of the teacher phase)

\[
X_{j,P,i} = X_{j,P,i} + r \times (X_{j,P,i} - X_{j,Q,i}) \tag{7}
\]

\[
X_{j,P,i} = X_{j,P,i} + r \times (X_{j,P,i} - X_{j,Q,i}) \tag{8}
\]

Accept \( X_{j,P,i} \) if it gives a better function value.

III. IMPLEMENTATION OF TLBO FOR TRAINING THE NEURAL NETWORKS

In this work, We propose a feed forward multilayered perceptron neural network (MLP-NN) to three layers, input layer with \( N_e \) neurons (geometry parameter of structure), hidden layer with \( N_c \) neurons and output layer with \( N_s \) neurons (physical parameter). The connection weight from the neurons of input layer to the neurons of hidden layer is \( WE \) and the connection weight from the neurons of hidden layer to the neurons of output layer is \( WS \) Fig. 1. The training of neural networks is to find an algorithm for optimized weights of networks to minimize the mean square error (MSE) described as follows

\[
MSE = \frac{(1/PT) \sum_{r=1}^{PT} \sum_{k=1}^{N_c} (Y_3 - Y) \times (Y_3 - Y)}{2} \tag{9}
\]

Where \( PT \) the total number of training samples, \( Y_3 \) is the output of the network and \( Y \) is the desired output.

\[
Y_3 = f_3 \left( \sum_{k=1}^{N_s} WS \times f_1 \left( \sum_{k=1}^{N_c} WE \times X \right) \right) \tag{10}
\]

With \( f_3 \) and \( f_1 \) are the activation functions (typically: sigmoid, tanh ...). \( X \) is the input vector of FNN.

![Feed-forward neural network architecture](image)

Fig. 1. Feed-forward neural network architecture

The implementation steps of the TLBO for training the NN are given in this section

Step 1. Define the neural network architecture: number of neurons in input layer \( N_e \), number of neurons in hidden layer \( N_c \) and number of neurons in output layer \( N_s \).

Step 2. Initialize the optimization parameters and define the optimization problem:

- Population size (number of learners in a class) \( N_p \)
- Number of generations (maximum number of allowable iterations) $maxit$
- Design variables of the optimization problem (i.e. number of subjects offered to the learner) $WE$ and $WS$ the matrices of input connection weights and output connection weights respectively. $WE$ matrix of $N_r$ rows and $Ne$ columns and $WS$ matrix of $Ns$ rows and $Nc$ columns.
- Define the optimization problem (fitness function): find the optimal $WE$ and $WS$ which minimizes the mean square error (MSE) Equation (9).

Step 3. Initialize the population according to the population size and the number of neurons and evaluate the corresponding objective function value. For simplification, the population is decomposed into two groups one represents the inputs weights population $WEP$ and the second one represents the output weights population $WSp$.

Step 4. The teacher phase starts by identifying the teacher (i.e. best solution) from the population. Then calculation of the mean result of each design variable.

Step 5. Evaluation of the difference between the current mean result and best mean result, according to Equation (5), the obtained difference is added to the current solution to update its values Equation (4). Accept of a new solution if it gives better function value. All the accepted function values at the end of the teacher phase are maintained and these values become the input to the learner phase.

Step 6. Update the learners’ knowledge by utilizing the knowledge of some other learner Equation (7) and (8). Accept of a new solution if it gives better function value. All the accepted function values at the end of the learner phase are maintained and these values become the input to the teacher phase of the next iteration.

Step 7. Repeat the procedure from step 4 to until the termination criterion is met.

IV. APPLICATION EXAMPLES AND RESULTS

In this section, the ability of neural networks trained with TLBO algorithm is assessed by applying it for the modeling of waveguide structures pseudo-elliptic waveguide filter Fig. 2 and H-plane waveguide filters considering rounded corners Fig. 3. The dimensions of the first filter [13] and second filters [14] are listed in Table 1 and Table 2 respectively.

The common parameters of algorithms (population size and number of iterations) and neural network parameters (Architecture, input parameter and their limit, the output parameters and frequency interval) of each structure are presented in Table 3. The activation functions are hyperbolic tangent function (Tansig), and linear function (Purelin) respectively. The other specific parameters of algorithms are given below.

PSO Settings: $c_1$ and $c_2$ are constant coefficients $c_1 = c_2 = 2$, $w$ is the inertia weight decreased linearly from 0.9 to 0.2.

TLBO Settings: for TLBO there is no such constant to set.

TABLE I. DIMENSIONS OF FIRST FILTER (UNITS: MILLIMETERS)

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>22.86</td>
</tr>
<tr>
<td>$b$</td>
<td>10.16</td>
</tr>
<tr>
<td>$l_1$</td>
<td>13.18</td>
</tr>
<tr>
<td>$l_2$</td>
<td>21.43</td>
</tr>
<tr>
<td>$l_3$</td>
<td>15.39</td>
</tr>
<tr>
<td>$l_4$</td>
<td>15.56</td>
</tr>
<tr>
<td>$l_5$</td>
<td>20.43</td>
</tr>
<tr>
<td>$l_6$</td>
<td>14.31</td>
</tr>
<tr>
<td>$v_1$</td>
<td>4.78</td>
</tr>
<tr>
<td>$v_2$</td>
<td>6.86</td>
</tr>
<tr>
<td>$h_1$</td>
<td>3.10</td>
</tr>
<tr>
<td>$h_2$</td>
<td>3.28</td>
</tr>
<tr>
<td>$W_1$</td>
<td>11.37</td>
</tr>
<tr>
<td>$W_2$</td>
<td>9.93</td>
</tr>
<tr>
<td>$W_3$</td>
<td>11.04</td>
</tr>
</tbody>
</table>

TABLE II. DIMENSIONS OF SECOND FILTER (UNITS: MILLIMETERS)

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>34.85</td>
</tr>
<tr>
<td>$b$</td>
<td>15.79</td>
</tr>
<tr>
<td>$t$</td>
<td>1.88</td>
</tr>
<tr>
<td>$l_1$</td>
<td>22.88</td>
</tr>
<tr>
<td>$l_2$</td>
<td>25.33</td>
</tr>
<tr>
<td>$l_3$</td>
<td>25.57</td>
</tr>
<tr>
<td>$W_1$</td>
<td>15.10</td>
</tr>
<tr>
<td>$W_2$</td>
<td>9.03</td>
</tr>
<tr>
<td>$W_3$</td>
<td>7.98</td>
</tr>
</tbody>
</table>
TABLE III. COMMON PARAMETERS AND NEURAL NETWORKS PARAMETERS

<table>
<thead>
<tr>
<th>Structure</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>3-8-40</td>
<td>3-11-54</td>
</tr>
<tr>
<td>Input neurons</td>
<td>W1, W2, W3</td>
<td>W1, W2, W3</td>
</tr>
<tr>
<td>Input interval(mm)</td>
<td>10.23&lt;W1&lt;12.51</td>
<td>13.95&lt;W1&lt;16.61</td>
</tr>
<tr>
<td></td>
<td>8.94&lt;W2&lt;10.92</td>
<td>8.13&lt;W2&lt;9.93</td>
</tr>
<tr>
<td></td>
<td>9.94&lt;W3&lt;12.14</td>
<td>7.182~W3&lt;8.78</td>
</tr>
<tr>
<td>Output neurons</td>
<td>S11(f)</td>
<td>S11(f)</td>
</tr>
<tr>
<td>Frequency interval(GHz)</td>
<td>[8 - 12]</td>
<td>[6.6 – 7.4]</td>
</tr>
<tr>
<td>Maxit</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>NP</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

Fig.4 shows the convergence of TLBO and PSO algorithms for minimizing the MSE of neural networks for two filters above-mentioned.

Fig.4. Convergence of algorithms for minimizing the MSE, a) first filter, b) second filter.

It is observed from Fig.4 that, the TLBO algorithm performs better in terms of convergence than the PSO algorithm, in which TLBO algorithm converge to 0.0059 and PSO algorithm converge to 0.1403 in 500 iterations for the first filter, and for the second filter TLBO algorithm converge to 0.0051 and PSO algorithm converge to 0.1363 in 1000 iterations.

Fig.5 gives a comparison between the reflection coefficient responses obtained by using the TLBO neural networks training and the references, an excellent approximate can be observed.

Fig.5. A comparison between the reflection coefficient responses obtained by TLBO-NN and the references, a) first filter, b) second filter

V. CONCLUSION

In this paper, Teaching-Learning-Based Optimization (TLBO) is proposed for training and testing feed-forward neural networks (FNN) for modeling waveguide filter structures (Pseudo-Elliptic waveguide filter and H-plane waveguide filters considering rounded corners). The results show the efficiency of TLBO algorithm, where TLBO algorithm converges to global minimum faster than PSO algorithm. The main advantage of this algorithm does not require selection of the algorithm-specific parameters.
References


