NEURAL NETWORK BASED LOAD FLOW ANALYSIS TO RADIAL DISTRIBUTION NETWORKS

M.Suresh, T.S.Sirish

1M.Tech.Student, 2Asso.Prof., Gayatri Vidya Prishad College of Engineering (Autonomous)

Abstract— load flow studies are performed to determine the distribution system static state at each node to find the steady state operating condition of a distribution system. Load flow study is the most frequently carried out study used by distribution companies for all the stages of system planning, operation and control. In this paper Radial Bias Function Networks (RBFN) and feed forward network with back propagation algorithm are proposed to solve load flow problem under different loadings conditions for computing node voltage magnitudes of the distribution system. The off-line load flow calculation is carried out by network topology based three phase distribution load flow. The results of network topology based three phase distribution load flow calculations are used to train the Radial Bias Function and feed forward network with back propagation algorithm.

Keywords—Back Propagation Algorithm (BPA), Load flow solution, RBFN, Mean Square Error (MSE)

I. INTRODUCTION

Distribution load flow is a very important and efficient tool for the analysis of distribution systems during many real time applications such as planning, operation and control. The conventional load flow methods Gauss Seidel (GS) and Newton Raphson (NR) method do not converge for distribution systems. Neural network works are widely used in power sector for different applications like optimization, protection, fault classification and load forecasting. The objective of neural network based load flow analysis is to reduce the computational time. A network topology based three phase distribution load flow is used for off-line load flow calculation and the data obtained from this load flow solution is used to train Radial Bias Function neural network and feed forward neural network with back propagation algorithm. Neural networks are most efficient to describe internal relationship with in the raw data not explicitly given, and there is no need to adopt any linear relationship between data.

II. POWER FLOW SOLUTION:

The study of various methods of solutions for power system network is referred as load flow study. This solution gives the voltages at various nodes, power flow through various lines and losses. The load flow studies are conducted for the purpose of system planning (short, medium and long) stages, operation and control. For the purpose of load flow solution, it is assumed that three phase balanced distribution system and the variables associated with each node are voltage, active power, and reactive power and load angle.

2.1. Three Phase Topology Based Distribution Load Flow Solution:

Algorithm for Distribution System Load Flow:

The algorithm steps for load flow solution of distribution system are given below:

Step 1: Read the distribution system line data and load data.

Step 2: Form the BBC matrix, the relationship can be expressed as

\[ [B] = [BBC][I] \] (3.1)

Step 3: Form the BCBV matrix, the relationship can be expressed as

\[ [\Delta V] = [BCBV][I] \] (3.2)

Step 4: Form the DLF matrix by using the eq. (3.1). The relationship will be

\[ DLF = [BCBV][IBC] \] (3.3)

\[ [\Delta V] = [DLF][I] \] (3.4)

Step 5: Set Iteration k = 0

Step 6: Iteration k = k + 1.

Step 7: Update voltages by following eq., as

\[ V^k = V^k + \Delta V \] (3.5)

\[ [\Delta V] = [DLF][I] \] (3.6)

\[ [\Delta V] = [V^k + \Delta V] \] (3.7)

Step 8: If max \[ |[\Delta V]| \] > Tolerance go to step 6.

Step 9: Calculate line flows, and losses from final bus voltages.

Step 10: Print bus voltages, line flows and losses.

Step 11: Stop

III. RADIAL BIAS NEURAL NETWORK FUNCTION:

Due to simplicity in structure this method becomes more popular.

![Fig1: Structure of radial bias neural network function](image)
The input neurons are directly connected to the hidden layer neurons. The output of the jth hidden neuron can be written as:

\[ h_j = \frac{\phi(\|x - c_j\|)}{\delta_j} \]

Where \( h_j \) is the output of the jth neuron, \( \phi \) is the nonlinear radial basis function, \( x \) is the input vector, \( c_j \) is the neurons centre and \( \delta_j \) is the centre spread parameter. The non-linearity function of RBFN is Gaussian function in following:

\[ h_j = \exp\left(-\frac{(x - c_j)^2}{\delta_j^2}\right) \]

The neurons of the output layer have a linear transfer function. Usually the linear transfer is \( \phi \). For the Kth neuron the output \( y_k \) is

\[ y_k = \sum_{j=1}^{m} w_{jk} h_j \]

Where \( h_j \) is the synaptic weight connecting hidden neuron j to output neuron k and m is the number of hidden layer neurons

IV. FEED FORWARD NETWORK WITH ERROR BACK PROPAGATION ALGORITHM:

Among the artificial neural networks, feed forward network with back-propagation algorithm are being extensively used for power system problems. For this application, a multi-layer feed forward network with error back-propagation has been employed. The load flow data that are used as training sets are same as that of Radial Bias Function neural network.

4.1. Back-Propagation Algorithm:

The Back-propagation algorithm (BPA) is also known as the generalized delta rule. The algorithm provides a method of calculating the gradient of the error function through the use of differentiation. As the name implies, the error is propagated backward from the output, one layer at a time. The first step towards the implementation of the back propagation procedure involves the forward propagation of training examples through the Network similar to the Perceptron. These training data are fed to the Network layer by layer. At each layer, activation potential is determined. Forward propagation to done to calculate cell activation. The error after initial computation in the forward pass is propagated from the output layer, layer by layer to the front. Repetition of the process will be done till a relatively small rate of change of error is achieved.

Algorithm
1. Select the training pair from the training set.
2. Apply the input vector to the network input neurons.
3. Calculate the output of the network.
4. Calculate the error between the network output and desired output (the target vector from the training pair)
5. Adjust the weights of the network in a way that minimizes the error.
6. Repeat steps (1) through (4) for each vector in the training set until the error for the entire set is acceptably low.

Step wise Mathematics of BPA
- Input: \( X_1, X_2, \ldots, X_n \)
- Output: \( Y \)
- Model: \( Y = f(X_1, X_2, X_3) \)
- Given the Architecture there are \( n \) weights to decide in a random fashion.
- \( W = (W_1, W_2, \ldots, W_n) \).
- Training Data: \((Y_1, X_{11}, X_{12}, \ldots, X_{1n})\) \(i= 1, 2, \ldots, n\)
- Given a particular choice of \( W \), we will get predicted \( Y \)’s
- \(( V_1, V_2, \ldots, V_n) \).
- Choose \( W \) such that over all prediction error \( E \) is minimized

\[ E = \sum (Y_j - V_j)^2 \]

4.2. Weight adjustment formula in Back Propagation

There are several weight adjustment formulas in the study of neural networks but the one used widely gradient descent method, which calculates the weights to the closest accuracy in order to find the results from the output neurons.

Gradient Descent Method

For every individual weight \( W_i \), updating formula looks like

\[
\frac{W_{\text{new}}}{W_{\text{old}}} = \frac{W_{\text{old}}}{W_{\text{old}}} + \alpha \times \left( \frac{\partial E}{\partial W} \right) \\
\alpha = \text{Learning Parameter (between 0 and 1)}
\]
Another slight variation is also used sometimes

\[ W_{r+1} = W_r + \frac{\partial E}{\partial W_r} + \beta \times (W_r - W_{r-1}) \]

\( \beta = \text{Momentum (between 0 and 1)} \)

V. MATLAB/SIMULATION RESULTS:

Matlab simulation results have been carried for 15-node system and 33-node systems. Network topology based three phase distribution load flow is used for generate training patterns for Radial bias Function neural network and multi-layer feed forward neural network. By using the off-line load flow there are 226 cases different of loadings of voltages have been generated for 15-node system.

**Type1:** The loadings are varied from 10% to 160% simultaneously at all nodes (16-case).

**Type2:** 210 cases of loadings are varied from 10% to 150% at each and every node keep as the remaining nodes corresponding loads are constant. Similarly there are 496 cases of loading patterns have been generated for 33-node system. Variation of active and reactive powers used to train the neural networks both Radial bias function neural network and multi-layer feed forward neural network with back propagation algorithm and corresponding voltages are set as targets to the networks. Neural network based load flow data is more accurate and also less computational when compared with existing load flow methods. Testing data has been taken as data not present in earlier training data and mean square error is very less. Some of the training data has been used for testing the error is approximately zero for RBFN.

<table>
<thead>
<tr>
<th>system</th>
<th>Voltage error(mse)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training data</td>
</tr>
<tr>
<td>15-node</td>
<td>0</td>
</tr>
<tr>
<td>33-node</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table1. Test results for RBFN.**

<table>
<thead>
<tr>
<th>System</th>
<th>Voltage error(mse)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training data</td>
</tr>
<tr>
<td>15-node</td>
<td>0.00030</td>
</tr>
<tr>
<td>33-node</td>
<td>0.00040</td>
</tr>
</tbody>
</table>

**Table2. Test results for multilayer feed forward neural network.**

<table>
<thead>
<tr>
<th>methods</th>
<th>Computational time(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed method</td>
<td>0.004</td>
</tr>
<tr>
<td>D. Das [5]</td>
<td>1.90</td>
</tr>
<tr>
<td>S. Ghosh and D. Das [6]</td>
<td>1.41</td>
</tr>
<tr>
<td>Ranjan and D. Das [7]</td>
<td>1.59</td>
</tr>
<tr>
<td>JIN-HAO TENG</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Table3. Comparison of proposed method with existing methods.

**CONCLUSION:**

In this paper an efficient and intelligent method for load flow solution to radial distribution networks has been proposed. By comparing the results of the load flow solutions by RBFN network and feed forward network with back propagation algorithm gives accurate results in acceptable range. In testing the neural network RBFN zero mean square error for trained data. But feed forward network gives some small error although RBFN gives zero error feed forward network with back propagation algorithm gives less error than RBFN other than trained testing patterns. Finally network with back propagation gives more accuracy than RBFN and also less computational time when compared with other existing load flow solutions to the radial distribution systems.

**REFERENCES**


