Fast Line Flows Estimation Using Parallel Self-Organising Hierarchical Neural Network

Laxmi Srivastava  
Department of Electrical Engg  
M.I.T.S., Gwalior  
Gwalior-474 005, INDIA

S.N. Singh & J. Sharma  
Department of Electrical Engg  
University of Roorkee,  
Roorkee-247 677, INDIA

Abstract: Fast MW security (line security) monitoring and analysis have assumed importance for power system static security analysis and fast prediction of branch flows is essential for this. An approach based on parallel self-organising hierarchical neural networks is presented to predict line flows in an efficient manner. Parallel self-organising hierarchical neural networks (PSHNN) are multistage networks in which stages operate in parallel rather than in series during testing. The entropy concept has been used to select the inputs for PSHNN. A revised back propagation algorithm is used for learning input non-linearities, along with forward-backward training. The proposed method is tested on IEEE 30-bus system to predict line-flows at different loading conditions.

I. INTRODUCTION

In steady-state security assessment of a power system, it is important to predict the line flows and bus voltages for different loading and network conditions of a power system [1,2]. In the literature, several approaches such as DC power flows [2], the P-1Q iteration method, distribution factor [3], the bounding method etc. have been proposed to estimate line flows for real time applications which are, in general, less accurate.

With the development of artificial intelligence and artificial neural networks (ANN) in recent years there is growing interest for proposing these tools to different areas of power system [4-7]. In [8], artificial neural network was used for dynamic security assessment of power system. The Multi Layer Perceptron (MLP) model with back propagation algorithm (BP) was applied to predict the critical clearing time (CCT) under different operating conditions and topology of the power system. Hsu et. al [9] developed a fast voltage estimation method using four-layered ANN. In the design of ANN, sets of variables that affect bus voltage most were selected as the input to the ANN using an entropy function. Ghosh et. al. [10] designed a multi-layer neural network for line flow contingency ranking. A regression-based correlation technique was used for feature selection to train the ANN using back propagation algorithm. Though quite a few researchers have used neural network for line security, however no attempt has been made in applying ANN for line-flows estimation.

In this paper, for accurate estimation of line-flows supervised learning has been applied to artificial neural networks. A separate PSHNN has been trained to predict line flow in each line. The PSHNN [11,12] has many desirable properties, such as optimized system complexity, minimized learning and recall times and truly parallel architecture in which all stages operate simultaneously without waiting for data from each other during testing. The inputs to all the stage neural networks are the system variables (real and reactive power injections) that affect the line-flows most. An approach based on system entropy [12,13] has been used to determine the input features for
the PSHNN. The proposed method is tested on IEEE 30-bus system.

II. METHODOLOGY

The general block diagram of the work carried out in this paper is presented in Fig. 1. A large number of patterns are generated in wide range of load variations randomly at each bus. The input features are selected to reduce the dimensionality of the input as well as size of the neural network, using entropy concept (block I). Input data thus generated are normalised (block II). For each line, supervised learning has been applied for accurate estimation of line-flows using parallel self-organising hierarchical neural network (block III). The block diagram of PSHNN used in the present work is shown in Fig. 2. The PSHNN consists of four stage neural networks (SNN). Each SNN is a three-layered revised back propagation (RBP) network having linear input & output units and nonlinear hidden units.

![Fig. 2 Block Diagram for a four-stage PSHNN](image)

The RBP algorithm is used to train each SNN in two steps. During step 1, it is same as the usual back propagation algorithm [14]. In the II step, the weights between the input layer and the hidden layer are fixed, only the weights between the hidden and output layers are retained. After first stage neural network (SNN1) is trained using RBP algorithm, the error signal of SNN1 is considered as the desired output for the next stage neural network (SNN2) and the weights are updated accordingly. The RBP algorithm is identically applied to all the four stages, which constitutes one sweep. The training of PSHNN is continued for a number of sweeps till convergence is obtained. For faster learning the forward-backward training is also adopted.

A. Entropy Concept

If a large number of inputs are used in an ANN, the number of nodes and thus number of interconnection weights will increase and the training of neural network will be extremely slow. To overcome this problem, only those variables that have significant effect on the line flow are selected as input to the neural network. An approach based on system entropy [9,12,13] has been used to select the features (i.e. real & reactive power injections affecting a line flow most).
The term entropy has been used to describe the degree of uncertainty about an event. A large value of entropy indicates high degree of uncertainty and minimum information about an event. It is observed that additional information about an event results in reduction of entropy value. The change in entropy for given information is defined as the information gain or entropy gain. This information gain provides a basis for feature selection.

Let an event be defined by a discrete random variable $S_i$ ($i = 1, 2, \ldots, q$) with $q$ possible values and let their respective probability function be $\text{prob} (S = S_i) = P_i$. Then the entropy function to the event $S$ is defined as

$$H(S) = \sum_{i=1}^{q} P_i \ln \left( \frac{1}{P_i} \right)$$

(1)

A large value of entropy indicates high degree of uncertainty and minimum information about an event. The entropy for a fair die with equal probability of occurrence for each number is given by

$$H_1 = 6 \times \frac{1}{6} \ln 6 = 1.792$$

(2)

If we do not have any information about the die except that it is fair, i.e. all events have equal probability to appear in the trial, the entropy will be high. Suppose on experimentation it is found, that the die is unfair and probabilities of occurrence for these numbers are 0.1, 0.1, 0.1, 0.1, 0.1 and 0.5. The entropy now become

$$H_2 = 5 \times 0.1 \ln 10 + 0.5 \ln 2 = 1.4898$$

It is observed that this additional information results in reduction of entropy value. The change in entropy for given information $S_1$ (fair die) and $S_2$ (information about the die from trials) is defined as the information gain or entropy gain ($G$)

$$G = H_1 - H_2$$

$$= \sum_{i=1}^{q} P_{1i} \ln \left( \frac{1}{P_{1i}} \right) - \sum_{i=1}^{q} P_{2i} \ln \left( \frac{1}{P_{2i}} \right)$$

(3)

Since, sum of the probabilities of all the events is unity, the above equation of information gain may be rewritten as

$$G = \sum_{i=1}^{q} P_{1i} \ln \left( \frac{P_{1i}}{P_{2i}} \right)$$

(4)

This information gain provides a basis for feature selection. The features are selected on the basis of maximum information gain provided by various trials.

B. Feature Selection

A large number of load patterns has been generated in a wide range of system operating conditions and AC load flow has been performed to obtain the line-flows for each case. The information gain is computed by observing the line-flow at each line for load disturbance at various buses in the power system and, on this basis, the input features (i.e. the real and reactive power injections affecting a line-flow most) are selected for PSHNN.

III. SOLUTION ALGORITHM

The detailed stepwise solution algorithm is given below:

(i) A large number of load patterns are generated randomly in a wide range of load variation at each bus.

(ii) AC load flow is performed for each load pattern to calculate line-flows.

(iii) Input features ($P_i$ and $Q_i$) are selected on the basis of entropy gain.

(iv) Input data are normalized and a PSHNN consisting of four-stage neural networks (SNN) is designed to estimate line-flows at each line. The output of the $j$th unit of the $k$th layer is of the form

$$O_k(j) = f \left( \sum_{n=1}^{N} W_{k}(j,i) O_{k-1}(i) \right)$$
(v) After SNN1 is trained using RBP algorithm, the error signal is

\[ e_1(n) = P_{flo}(n) - O_1(n) \]  

(5)

(vi) Use the error signal \( e_1(n) \) as the desired output of SNN2 and \( s'(n) \) as the input vector to train SNN2. The error signal for the second stage is

\[ e_2(n) = e_1(n) - o_2(n) \]  

(6)

(vii) The same procedure is adapted to train SNN3 & SNN4.

The final output of PSHNN is

\[ P_{flo}(n) = O_1(n) + O_2(n) + O_3(n) + O_4(n) \]  

(7)

(viii) The RBP algorithm is identically applied to all the four stages. During step I of RBP algorithm, the sum of squared error is minimized by generalized Delta rule given by

\[ E_i = \frac{1}{2} \sum_{n=1}^{N} \left[ P_{flo}(n) - O_i(n) \right]^2 \]  

(8)

(ix) The connection weights are updated using equations

\[ \Delta W_k(j,i) = \eta. \delta. O_i(n) \]  

(9)

For output nodes

\[ \delta_j = \sum_{n=1}^{N} f'(\cdot) \left[ P_{flo}(n) - O_i(n) \right] \]  

(10)

For hidden nodes

\[ \delta_j = \sum_{n=1}^{N} f'(\cdot) \left[ \sum_i \delta_i W_k(j,i) \right] \]  

(11)

(x) In the II-step, the weights between the output and hidden layer are retrained using (9) and (10).

(xi) The iterations are continued until the error becomes negligible.

(xii) The same procedure from step (ix) to (x) is adapted for the succeeding stages and the final error signal of the PSHNN becomes

\[ e_f(n) = P_{flo}(n) - P_{flo}(n) \]  

(12)

(xiii) After all the four SNNs are trained, the retraining of SNN3 and SNN2 is performed. This constitutes one sweep and referred to as forward-backward training.

(xiv) Training of the PSHNN (step (vi) to step (xiii)) is continued for a number of sweeps until the convergence is obtained.

IV. RESULTS AND DISCUSSIONS

The proposed method has been tested on the IEEE 30-bus system [15]. For this system, 1000 load patterns were generated randomly. Out of which, 800 patterns were used for training purposes and remaining 200 patterns were used for the testing. The 14 input features were selected for the test systems based on entropy gain to train and test the PSHNN. The root mean square (rms) error of line flow estimation has been calculated as

\[ RMS \ Error = \sqrt{\frac{1}{P} \sum_{i=1}^{P} \left[ T(i) - O(i) \right]^2} \]  

(13)

Where \( P \) is the number of patterns, \( T(i) \) is the target output and \( O(i) \) is the actual output of ANN for \( i^{th} \) pattern.

IEEE 30-Bus System

Since the IEEE 30-bus system [15] has 6 PV buses, 24 PQ type buses and 41 lines, a separate neural networks are required to be trained to predict all the 41 line-flows under changing operating conditions. The load patterns were generated randomly by perturbing the load on each bus in the range of 50% to 150% of the base case. The input data were normalized between 0.1 and 0.9.

Considering each line separately, a four-stage PSHNN was developed. The PSHNN has 14 neurons in the input layer, five neurons in hidden layer and one neuron in the output layer (14-5-1), in addition to one auxiliary neuron, each, in the input and the hidden layer, to predict the desired line-flows. The line-flows of each line were
predicted with inputs selected by entropy reduction approach. The selected input features for prediction of line-flows in some of the lines are given in Table 1.

From Table 1 it is observed that for estimation of line-flows in lines 27 and 38, even, the real and reactive power injections at the terminal buses of the lines are not selected as the input features for the PSHNN obtained using entropy concept.

Estimated line-flows of line –14 for 15 test patterns only are shown in Table 2 due to limited space. Summary of line-flows prediction error in lines nos. 14, 21, 28 and 39 are presented in Table 3.

<table>
<thead>
<tr>
<th>Pattern no.</th>
<th>Actual flow</th>
<th>Output flow</th>
<th>Max. absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.28524</td>
<td>0.28632</td>
<td>-0.00108</td>
</tr>
<tr>
<td>2</td>
<td>0.26788</td>
<td>0.26702</td>
<td>0.00086</td>
</tr>
<tr>
<td>3</td>
<td>0.31931</td>
<td>0.31787</td>
<td>0.00144</td>
</tr>
<tr>
<td>4</td>
<td>0.25829</td>
<td>0.25892</td>
<td>-0.00063</td>
</tr>
<tr>
<td>5</td>
<td>0.27458</td>
<td>0.27496</td>
<td>-0.00038</td>
</tr>
<tr>
<td>6</td>
<td>0.29553</td>
<td>0.29503</td>
<td>0.00050</td>
</tr>
<tr>
<td>7</td>
<td>0.30454</td>
<td>0.30495</td>
<td>-0.00041</td>
</tr>
<tr>
<td>8</td>
<td>0.33716</td>
<td>0.33891</td>
<td>-0.00175</td>
</tr>
<tr>
<td>9</td>
<td>0.32775</td>
<td>0.32507</td>
<td>0.00268</td>
</tr>
<tr>
<td>10</td>
<td>0.24594</td>
<td>0.24569</td>
<td>0.00025</td>
</tr>
<tr>
<td>11</td>
<td>0.34131</td>
<td>0.34375</td>
<td>-0.00244</td>
</tr>
<tr>
<td>12</td>
<td>0.23812</td>
<td>0.23895</td>
<td>-0.00083</td>
</tr>
<tr>
<td>13</td>
<td>0.25745</td>
<td>0.25542</td>
<td>0.00203</td>
</tr>
<tr>
<td>14</td>
<td>0.26523</td>
<td>0.26411</td>
<td>0.00112</td>
</tr>
<tr>
<td>15</td>
<td>0.30415</td>
<td>0.30496</td>
<td>-0.00081</td>
</tr>
</tbody>
</table>

From Tables 2 and 3, it is observed that the maximum absolute error in line-flows prediction is 0.00353, 0.00254, 0.00381, and 0.00278pu at lines 14, 21, 28 and 39 respectively. The rms errors at these lines are 0.00066, 0.00083, 0.00078 and 0.00084pu respectively. Thus, the proposed model accurately predicts the line flow for this system under changing load condition.

V. CONCLUSIONS

An approach based on parallel self-organising hierarchical neural network has been developed to estimate line-flows in an efficient manner. To reduce the training
time and enhance the accuracy of the PSHNN, four stage neural-networks were employed in PSHNN, and inputs to the neural network were selected on the basis of information gain. The designed PSHNN has been applied to predict line-flows under changing operating condition of the power system. Once the PSHNN is trained, it predicts quick results for unknown load patterns, as all the four stages operate in parallel during testing. Test results of the sample system reveal that maximum absolute error is very small.

The computation of line-flows by load flow analysis takes long time as it should be run for any change in load/generation. On the other hand, by the proposed method, once the training of the PSHNN is successfully completed, the prediction of the line-flows is almost instantaneous. This can be used for real time application.

VI. REFERENCES


