Elman Neural Network Backpropagation Based Evaluation of Critical Busbars in Power Systems with Renewable Sources

P K Sahoo[*], P K Satpathy[**‡], M N Mohanty[***]

[*]Dept. of Elect. Engg., ITER, S'o'A University, Bhubaneswar, Odisha, India

[**]Dept. of Electrical Engg., CET, Bhubaneswar, Odisha, India

[***]ITER, S'o'A University, Bhubaneswar, Odisha, India

(pradyumnakumar_sahoo@yahoo.co.in, satpathy.pks@gmail.com, mihir.n.mohanty@gmail.com)

[‡] Department of Electrical Engineering, College of Engineering and Technology,
Ghatikia, Bhubaneswar, Odisha, PIN-751003, India, Tel: +91 674 239 1319,
Fax: +91 674 239 1324, satpathy.pks@gmail.com

Received: 05.02.2015 Accepted:17.03.2015

Abstract- We present an efficient analysis for evaluation of critical busbars in electric power systems by judging the performance of various backpropagation schemes of Elman Neural Network. The objective of this study is to find the most efficient scheme that yields fastest convergence under supervised learning. The study is conducted on the standard IEEE 30-bus test system supplemented by renewable source of generation. Out of six backpropagation schemes tried in this work, it is observed that gradient descent backpropagation with momentum and adaptive learning rate performs exceedingly well by showing guaranteed convergence, which is relatively 1000 times faster than the conventional approach. It is also claimed that this study would be very helpful to the power system utilities and researchers in reducing the burden on the utility in terms of complexity and computational time requirements of conventional approach.

Keywords Voltage stability, critical busbars, neural network, backpropagation.

1. Introduction

Voltage instability, which often leads to ultimate collapse in electric power systems, poses several concerns for the power utilities [1-3]. The literature indicates that load buses having higher *L*-index are treated as critical ones for such studies [4]. Most of the earlier studies for identification of critical bus bars in a system are based on conventional approach. Such studies primarily depend on load flow simulation over a specific time period of observation and are subject to several limitations such as; complexity in modeling, unavailability of real time database, simulation of contingencies and more so. Hence, most of the present day research is inclined towards application of soft computing tools in power system analysis [5-11]. Neural network application for solving complex power system problems has become very popular among the researchers due to following reasons.

(i) It works as a black-box between the input space and target (output) space having lots of flexibility in functional mapping, which reduces the burden of strict mathematical formulation of complex systems.

(ii) It is capable of establishing faster correlations as compared to conventional methods for the sake of fitting a given function, pattern recognition, and data clustering through result classification.

These facts drive the authors of this paper to implement neural network tools for identification of critical busbars in electric power systems with renewable sources, through a comparative study and performance analysis. This paper explorers some of the practical limitations encountered by small hydro power (SHP) generating units and at the same time evaluates various possibilities so as to maintain continuous generation by these units while ensuring smooth evacuation of every unit of real power generated by such units to the neighboring grid in a grid connected power system scenario. In order to emphasize the merits of the proposed work, the proposed simulation has been tested under some worst system conditions considering the standard IEEE 30-bus test system by way of connecting the SHP unit to the most critical system bus of the test system through on-load tap changing transformer (OLTC) control of generator transformers at the local bus. The basic objective of this study is to evolve an optimal neural network structure and learning criteria that would result in faster convergence under variety of operating conditions. In the present context, the authors have successfully exploited the above mentioned attributes of neural networks by imparting thorough training till convergence is attained. The result of training convergence validates that a properly trained Elman neural network along with a suitable backpropagation learning rule can provide faster information on critical busbars in a voltage stability constrained power system.

In order to arrive at the stated goal, the authors have formulated an extensive database of occurrences happening in power system operation. This database is quite useful for preparation of input data and target data for the sake of imparting thorough training to the proposed neural network through backpropagation and supervisory learning mechanism. The authors have also presented a comparative performance analysis considering six forms of backpropagation schemes for this purpose, as indicated below.

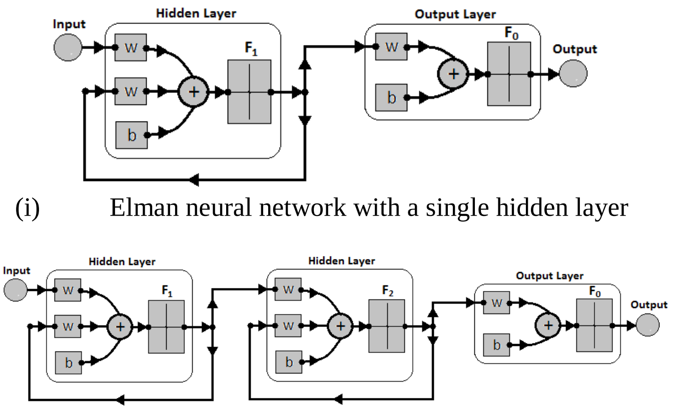
(i) *GD*: Gradient Descent backpropagation,
(ii) *GDM*: Gradient Descent backpropagation with Momentum,
(iii) *GDA*: Gradient Descent backpropagation with Adaptive learning rate,
(iv) *GDX*: Gradient Descent backpropagation with Momentum and Adaptive learning rate,
(v) *CGF*: Conjugate Gradient backpropagation with Fletcher-Reeves update, and
(vi) *SCG*: Scaled Conjugate Gradient backpropagation.

It is claimed in this work that application of an appropriate neural network is a better option as compared to conventional approaches, which significantly reduces the burden of repeat load flow execution for monitoring the critical busbars.

Section 2 of the paper contains the background of neural network structure with justification for its application to this study. In this section, multiple layers for the Elman neural network have been used for drawing a comparative performance analysis. In section 3, six types of backpropagation algorithms have been described for implementation in the neural network training. Section 4 highlights the methodology behind problem formation for conducting the proposed analysis. In Section 5, a detailed case study on the standard IEEE 30-bus test system is presented along with result analysis. The simulation results justify that the proposed algorithm works well in all situations with a minimum error in meeting the target during training convergence. The analysis shows that the Elman neural network supported with gradient descent backpropagation with momentum and adaptive learning rate performs convincingly well as compared to its counterparts in terms of iteration requirements and speed of convergence.

2. Historical Elman Neural Network

The literature shows that neural networks are capable enough in solving complex problems with ease and fastness [12-14]. It is also widely accepted in the power system area dealing with operation and control [15]. So many other researchers have also pioneered their quest for tracing further applications of neural networks in power systems [16-23]. However, the authors strongly feel that there might be plenty of further scopes to be explored in monitoring critical busbars in electric power transmission networks by applying various neural network schemes such as, feed forward backpropagation training rules, layer recurrent architecture, radial basis function, and historical Elman neural network techniques. In this paper, the historical Elman neural network architecture is considered. A generalized structure of the Elman neural network (with possible combinations of hidden layers) is presented in Fig.1.



(i) Elman neural network with a single hidden layer

(ii) Elman neural network with two hidden layers

Fig. 1. Structure of historical Elman neural network

The transfer functions for each hidden layer may differ from one another. In this paper, the authors have made use of three common types of transfer functions such as linear (*purelin*), tan-sigmoid (*tansig*), and log-sigmoid (logsig) as represented in Fig.2.

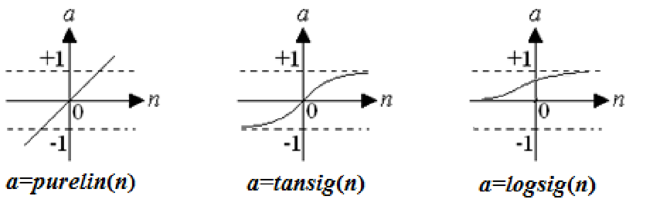
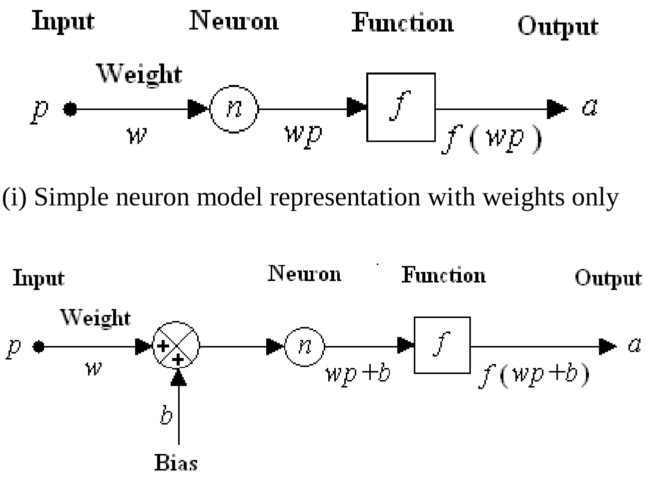


Fig. 2. Various transfer functions for Elman neural network

The simplified model is also shown in Fig.3 that illustrates the procedure adopted for adjustment of the weights (*w*) in the process of transformation of the input (*p*) into a corresponding output (*a*) with or without biasing (*b*). The mathematical expression of the model is presented in Eq. (1) and (2).

$$a = f(wp) \tag{1}$$

$$a = f(wp + b) \tag{2}$$



(i) Simple neuron model representation with weights only

(ii) Simple neuron model with weights and bias

Fig. 3. Neural Network input, output and transfer function

The Elman neural network makes use of feedback from the output of each hidden layer to the input of the corresponding layer that helps in detection and generation of time varying patterns. That is why the hidden layers are also treated as recurrent layers. The major difference between the proposed Elman network and the other types of static and dynamic neural networks (feed forward networks with backpropagation and layer recurrent networks) lies with the fact that Elman networks have the flexibility of assigning enough neurons in the hidden layers depending on the complexity of the problem. The Elman network also makes use of standard backpropagation techniques in order to resort to minimal error sequence during training.

3. Backpropagation Schemes for Elman Neural Network

This section presents six important types of backpropagation schemes, which have been considered in this work for drawing up the comparative analysis on training performance.

3.1 Gradient Descent (GD)

In this method of training the weights are adjusted considering the negative gradient of the path of the performance function. The gradient function is calculated from the derivative of the transfer function of the neurons in that layer. The mathematical expression for this operation is depicted in Eq. (3).

$$Z_{k+1} = Z_k - \alpha_k g_k \tag{3}$$

In this expression, Z_{k+1} refers to the recent weight and bias vector corresponding to the earlier values of the vector Z_k, gradient g_k, and the learning rate α_k.

3.2 Gradient Descent Momentum (GDM)

In this scheme an additional momentum function in the range of {0, 1} is added to the gradient descent training function discussed earlier. In this way the neural network enhances its capability to ignore minor changes in the gradient over the error surface and hence the training process becomes faster.

3.3 Gradient Descent with Adaptive Learning Rate (GDA)

This method is similar to the earlier method of gradient descent with momentum but has an additional feature of adaptive learning rate instead of a constant one. The adaptive learning rate enables the network to adapt to changes in the error surface during the training process by continuously monitoring the errors calculated during each epoch. While the recently calculated error exceeds the previous error by a specific pre-defined margin, repeat calculation of weights and bias is performed with a lower learning rate so as to maintain the stability.

3.4 Gradient Descent with Momentum and Adaptive Learning Rate (GDX)

This method is a combination of gradient descent with momentum and adaptive learning as well. This method is applicable to networks having derivative functions in its parameters such as weights, inputs, and transfer functions. The backpropagation scheme calculates the derivatives of performance function with respect to the weight and bias variables in order to adjust each variable according to gradient descent with momentum. In case of decreasing performance, the learning rate is increased by a given factor, else the learning rate is held constant.

3.5 Conjugate Gradient with Fletcher-Reeves Update (CGF)

In Fletcher-Reeves method, the technique of updating the weights and bias parameters is used as per the expressions shown in the Equations (4) through (6).

$$Z_{k+1} = Z_k + \alpha_k p_k \tag{4}$$

$$p_k = -g_k + \beta_k p_{k-1} \tag{5}$$

$$\beta_k = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} \tag{6}$$

Here, Z_{k+1} refers to the recent weight and bias vector corresponding to the earlier values of the vector Z_k, gradient g_k, and the learning rate α_k. The updated gradient may be calculated as per Equation (5) taking the current gradient g_k, a constant term β_k and the earlier gradient p_{k-1}. The procedure for calculation of β_k is given in Equation (6). This scheme adjusts the step size of the updates in every iteration by conducting a search along the conjugate gradient direction that helps in minimizing the performance function along that direction.

3.6 Scaled Conjugate Gradient (SCG)

It is often observed that conjugate gradient backpropagation scheme based on line search technique could be computationally expensive. In order to overcome this drawback the scaling of the gradient calculation may be helpful. In the scaled conjugate gradient method the scaling mechanism is incorporated by combining the model-trust region approach [24].

The performance study of an neural network to deal with random inputs requires a thorough analysis of output input correspondence (OIC), error surface (ES), training performance (TRP), training states (TRS), and receiver operating characteristic (ROC). While ‘OIC’ signifies the response of the output with respect to the inputs presented to the network, ‘ES’ refers to the error surface during training. The ‘OIC’ also justifies whether the proposed neural network is capable of producing valid outputs for all the inputs presented to it. On the other hand, the error surface displays, either the sum squared error during training convergence as a function of the weights and bias values in a three dimensional plane, or the error contour plot against the weights and bias values in a two dimensional plot, or both as well. The depression of the ‘ES’ shows the minimum error point that clearly indicates the suitable weights and bias values at that point.

Furthermore, ‘TRP’ and ‘TRS’ respectively highlight the path followed during each iteration (epoch) of training process and the error gradient along with the learning rate in the process of convergence subject to the limiting condition prevailing during training.

The ‘ROC’ quantifies the performance of the neural network by judging its ability to classify the outputs into specific categories in responce to the randomly presented input datasets. This plots makes use of threshold values, in the range of zero and unity to be applied on the output values, for computation and demonstration of variations in the false positive (FP) rate and true positive (TP) rate, as a function of the threshold. The expressions used for the computation of false positive rate (FPR) and true positive rate (TPR) are presented in Equations (7) and (8) respectively.

$$FPR = (NO_{LTT}/NT_0) \tag{7}$$

$$TPR = (NO_{GET}/NT_1) \tag{8}$$

Where,

NO_{LTT} = Number of output values less than threshold,

NO_{GET} = Number of output more than/same as threshold,

NT_0 = Number of targets with zero value, and

NT_1 = Number of targets with unity value.

In this work, the threshold is varied in the range {0, 1} with relatively slow variation (i.e. step interval of 0.1) that enables generation of more points in the FP-TP plane. If the ‘ROC’ curve stretches or stays inclined towards the upper-left corner in the corresponding FP-TP plotting frame, it implies perfect raining for the neural network.

4. Problem Formulation

In order to ensure reliable operation and smooth evacuation of power in case of grid connected SHPs, the major focus of study in this paper includes identification of following aspects.

Firstly, the standard IEEE-30 bus system as shown in Fig.4 is considered as the parent system representing the grid and the critical buses are identified prior to the connection of the SHP. This analysis is based on application of the Elman neural network. The training for the neural network is performed through six models of backpropagation schemes. Then, the training performance of these six schemes are compared with each other in terms of (i) number of training cycle requirement and (ii) time consumtion for successful convergence of training, in order to identify the best training condition.

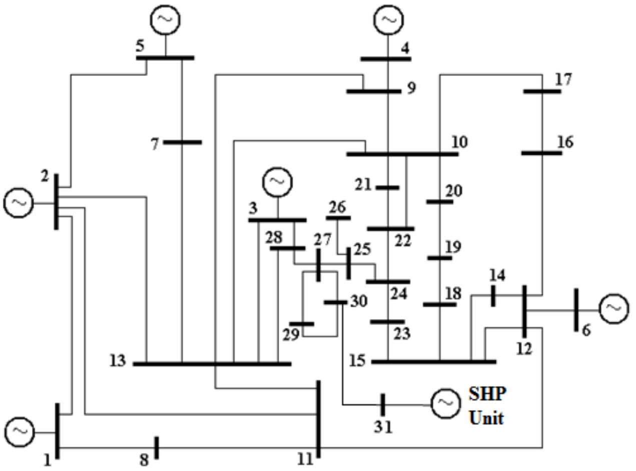


Fig. 4. IEEE 30-bus system with Grid connected SHP unit

In the next assessment, the SHP unit is subsequently connected at one of the buses of the parent system (called the pulling bus, i.e. bus-30), in order to study the possibility of any change occurring in the location of the critical bus during power evacuation by the SHP to the grid.

In actual geographic conditions, the SHP units are situated at remote locations that may be far off from the grid structure. In order to connect them to the grid, often a secondary feeder may be essential. This feeder acts as the link or interconnection between the SHP and a specific bus point of the grid usually denoted as the evacuation point. As long as the evacuation point remains strong and stable, the evacuation is made safely. However, the real problem mounts if the evacuation point happens to be a weak bus in the system.

It is of paramount importance to meet this challenge and ensure smooth evacuation at all times, which triggered the authors for conducting a study to identify the weakest bus in the system. The evacuation of power is done at the receiving bus, which is represented by bus-31 in Fig.4. However, it may be more accurate to mention at this stage that the SHP units may fail to support smooth power evacuation to the grid under some critical system conditions. Hence, the simulation requires a wide range of operating conditions (also known as situations or events) so as to generate an exhaustive input dataset for imparting successful training to the proposed neural network.

The simulation parameters considered in this paper for formation of the input dataset are illustrated in Table 1, which include initial voltage criteria, loading factor, and reactive power limit on the PV buses. In view of these parameters, the authors have considered a total of forty four situations for formation of the input dataset.

The next part of this work is focused at formulation of training sequences based on selection of neural network parameters (i.e. number of hidden layers and number of neurons in these layers) and the backpropagation schemes. In view of this a total of thirty six training sequences are formed (this includes nine sequences for networks with single hidden layer and twenty seven sequences for networks with two hidden layers) and specific training code (TC) is assigned for each of them, as shown in Table 2.

The input dataset matrix in the present case is of dimension [44×60]. Each row of the input matrix contains sixty elements comprising of thirty voltage magnitudes followed by thirty L-indices. The voltage magnitude and L-index are obtained by performing Newton-Raphson load flow solution of the IEEE 30-bus test system. On the other hand, each column of the input matrix contains forty four elements comprising the results for gradual loading conditions.

In this paper, the supervised learning rule has been followed to train the Elman neural network. The sequences of events that occur during each epoch of training cycle of an Elman neural network are as follows.

- In the beginning, the entire input dataset is presented to the network.
- Then, outputs are calculated and compared with the target to generate an error sequence.
- Next, the error is propagated backwards to find the error gradient for each weight and bias.
- This gradient is then used to update the weights for the next epoch.

The above data are further grouped into four different situations depending on initial voltage, loading scenario, and contingency criteria. In order to simulate the load growth in the system, the complex load of each bus has been increased progressively beyond base load in the system with the help of a loading parameter (λ) that gives the loading pattern 'S=(1+λ)(S_{base})'.

Also, it is often desirable that the input dataset be divided into three components, viz; training data (*TrD*), validation data (*VaD*), and test data (*TeD*) for better training performance. In this paper, the percentage ratio of input data division into these three categories is clearly illustrated in Fig.5 (i.e. *TrD* : *VaD* : *TeD* = 60 : 20 : 20).

The simulation of the proposed methodology is implemented on the basis of the flow chart of Fig.6. This flow chart demonstrates the steps and conditions followed in this work.

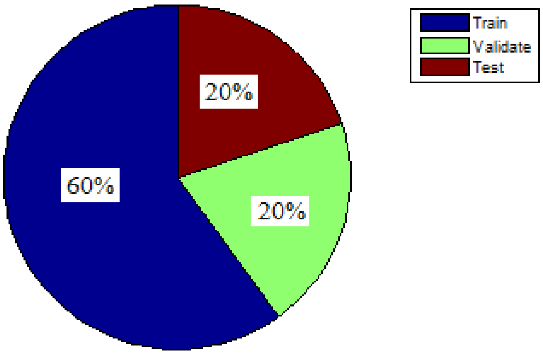


Fig. 5. Data division into Train, Validate, and Test data

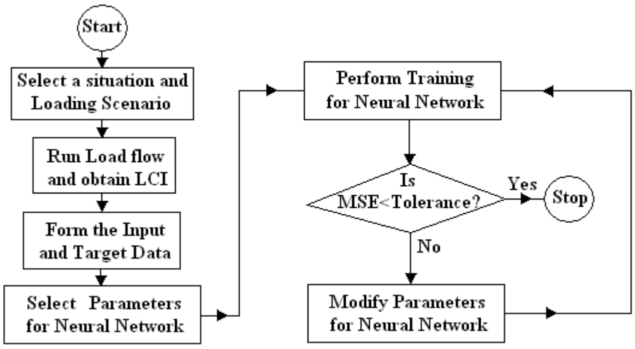

Start
Select a situation and Loading Scenario
Run Load flow and obtain LCI
Form the Input and Target Data
Select Parameters for Neural Network
Perform Training for Neural Network
Is MSE<Tolerance?
Yes
Stop
No
Modify Parameters for Neural Network

Fig. 6. Flow Chart for Neural Network Training

Table 1. Guidelines for formation of input dataset based on situation considered

Situation Nomenclature (SN)	Initial Voltage Criteria	Loading Factor (λ) *for S*=(1+λ)(S_{base})	Q_{lim} restriction on the PV buses
SN-1 to SN-11	Non uniform Specified Voltage	λ=0 to 1, step of 0.1	**No**
SN-12 to SN-22	Non uniform Specified Voltage	λ=0 to 1, step of 0.1	**Yes**
SN-23 to SN-33	Uniform voltage of flat 1p.u.	λ=0 to 1, step of 0.1	**No**
SN-34 to SN-44	Uniform voltage of flat 1p.u.	λ=0 to 1, step of 0.1	**Yes**

Table 2. Training code assignment based on transfer function allocation in the layers

Number of Hidden Layers (HL)	Transfer Function of first HL (F_1)	Transfer Function of second HL (F_2)	Transfer Function of output layer (F_0)	Training Code (TC)
1	Purely linear	---	Purely linear	11
1	Purely linear	---	Tan-sigmoid	12
1	Purely linear	---	Log-sigmoid	13
1	Tan-sigmoid	---	Purely linear	21
1	Tan-sigmoid	---	Tan-sigmoid	22
1	Tan-sigmoid	---	Log-sigmoid	23
1	Log-sigmoid	---	Purely linear	31
1	Log-sigmoid	---	Tan-sigmoid	32
1	Log-sigmoid	---	Log-sigmoid	33
2	Purely linear	Purely linear	Purely linear	111
2	Purely linear	Purely linear	Tan-sigmoid	112
2	Purely linear	Purely linear	Log-sigmoid	113
2	Purely linear	Tan-sigmoid	Purely linear	121
2	Purely linear	Tan-sigmoid	Tan-sigmoid	122
2	Purely linear	Tan-sigmoid	Log-sigmoid	123
2	Purely linear	Log-sigmoid	Purely linear	131
2	Purely linear	Log-sigmoid	Tan-sigmoid	132
2	Purely linear	Log-sigmoid	Log-sigmoid	133
2	Tan-sigmoid	Purely linear	Purely linear	211
2	Tan-sigmoid	Purely linear	Tan-sigmoid	212
2	Tan-sigmoid	Purely linear	Log-sigmoid	213
2	Tan-sigmoid	Tan-sigmoid	Purely linear	221
2	Tan-sigmoid	Tan-sigmoid	Tan-sigmoid	222
2	Tan-sigmoid	Tan-sigmoid	Log-sigmoid	223
2	Tan-sigmoid	Log-sigmoid	Purely linear	231
2	Tan-sigmoid	Log-sigmoid	Tan-sigmoid	232
2	Tan-sigmoid	Log-sigmoid	Log-sigmoid	233
2	Log-sigmoid	Purely linear	Purely linear	311
2	Log-sigmoid	Purely linear	Tan-sigmoid	312
2	Log-sigmoid	Purely linear	Log-sigmoid	313
2	Log-sigmoid	Tan-sigmoid	Purely linear	321
2	Log-sigmoid	Tan-sigmoid	Tan-sigmoid	322
2	Log-sigmoid	Tan-sigmoid	Log-sigmoid	323
2	Log-sigmoid	Log-sigmoid	Purely linear	331
2	Log-sigmoid	Log-sigmoid	Tan-sigmoid	332
2	Log-sigmoid	Log-sigmoid	Log-sigmoid	333

Table 3. Preliminary guiding results for training convergence

Parameters Assigned for the Neural Network				Training Results for Convergence	
Training Code (TC)	Epochs per training cycle	Hidden Layers	Neurons per Hidden Layer	Training Cycles undertaken	Time taken for convergence (s)
11	1000	1	5	Did not converge in 200 cycles	
11	100	1	5	35	99
11	10	1	5	15	8
111	1000	2	3	67	2206
111	100	2	3	38	133
111	10	2	3	20	13

Table 4. Final training results showing full/partial convergence in respective groups

Network Parameters# #(TC: Training Code, HL: Hidden Layers, NL: Neurons/ Layer)			**Convergence Results* for Various Backpropagation Schemes$** $(GD: Gradient Descent, GDM: Gradient Descent with Momentum, GDA: Gradient Descent with Adaptive learning, GDX: Gradient Descent with Momentum and Adaptive learning, CGF: Conjugate Gradient with Fletcher-Reeves update, SCG: Scaled Conjugate Gradient) *(CY: Training Cycles Performed for convergence, TI: Time for convergence in seconds, NC: No Convergence in as much as 200 training cycles)											
TC#	**HL#**	**NL#**	**GD$**		**GDM$**		**GDA$**		**GDX$**		**CGF$**		**SCG$**	
			CY*	**TI***	**CY***	**TI***	**CY***	**TI***	**CY***	**TI***	**CY***	**TI***	**CY***	**TI***
11	1	***4***	31	19	31	18	3	2	3	2	52	68	4	5
12	1	***5***	1	1	1	1	1	1	1	1	14	10	2	2
13	1	***5***	1	1	1	1	1	1	1	1	14	7	2	2
21	1	5	4	3	4	3	4	3	4	3	NC	--	NC	--
31	1	5	21	15	21	15	21	15	21	15	NC	--	NC	--
111	2	***4***	1	1	1	1	1	1	1	1	97	135	89	99
112	2	***5***	2	2	2	2	2	2	2	2	72	43	2	2
113	2	***5***	2	2	2	2	2	2	2	2	72	52	2	2
121	2	3	1	1	1	1	1	1	1	1	NC	--	NC	--
131	2	3	1	1	1	1	1	1	1	1	NC	--	NC	--
211	2	2	54	39	54	39	3	2	3	2	143	236	123	197
212	**2**	**4**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
213	2	***4***	1	1	1	1	1	1	1	1	135	92	1	1
221	2	3	4	4	4	4	4	4	4	3	NC	--	NC	--
231	2	3	4	4	4	4	4	4	4	3	NC	--	NC	--
311	2	2	24	19	54	42	3	2	3	2	NC	--	NC	--
312	2	***4***	1	1	1	1	1	1	1	1	141	138	23	27
313	2	***4***	1	1	1	1	1	1	1	1	11	8	1	1
321	2	3	4	4	4	4	4	4	4	3	NC	--	NC	--
331	2	3	4	4	4	4	4	4	4	3	NC	--	NC	--

5. Result Analysis and Discussion

The findings of the proposed work are presented here in a lucid way. It is observed from the conventional load flow simulation that bus-30 and bus-28 happen to be most critical buses of the IEEE 30-bus system alone (i.e. prior to the connect ion of the SHP). However, after the SHP is connected to bus-30, the location of the critical bus changes to bus-19 and bus-26. The simulation also indicates that the time elapsed in computing the critical bus for both the events (before and after connection of SHP) is around 946 seconds, which is considerably high. Therefore, the neural network methodology is applied to verify whether the computational time could be reduced.

In this section, we present the simulation results obtained from the training performance of the proposed Elman neural network. The results are presented, both in Table-3 and Table-4, in view of various situations of the test system (i.e. SN-1 to SN-44 of Table-1), parameters of the neural network (i.e. HL and TC values of Table-2; and NL values of Table-4), and six different backpropagation schemes (i.e. GD, GDM, GDA, GDX, CGF, and SCG schemes of Table-4). The basic objective of considering so many situations, parameters and backpropagation schemes is to find out the optimum combination of these items for obtaining best training performance within least computational time.

The proposed Elman neural network is trained by exposing it with a set of input data reflecting the trend and variations in voltage level at the bus bars of a power system subject to various practical situations. In order to accomplish this, the authors have judiciously selected the input data from Newton-Raphson load flow program for various scenarios as described in the preceding section. With these considerations, the proposed neural network training is performed on the IEEE 30-bus test system with grid connected SHP, by pursuing the following seven steps.

Step 1: *Formation of Input data (p)*: The sequential input data for the Elman neural network is formed with the help of guidelines described earlier in this section and Table 1. When arranged in matrix form, the input data matrix had dimension [44×60].

Step 2: *Formation of Target data (T)*: In this step, the sequential target data for the Elman neural network is formed with the help of guidelines described earlier in this section. When arranged in matrix form, the input data matrix had dimension [44×1].

Step 3: *Selection of neural network structure (f)*: In this case, the Elman neural network is selected with the provision for implementing six types of backpropagation training rules as described in section 3. Also, the performance analysis is carried out considering various combinations of hidden layers, number of neurons in respective layers and transfer function assignments as indicated in Table 2 and Table 4.

Step 4: *Initialization of neural network parameters*: In this step, the authors made use of following parameter selection for imparting training of the neural network.
Number of neurons per hidden layer: In the range {1, 5}.
Epochs per training cycle: 10 or 100 or 1000,
Error tolerance limit (ε): 0.001.

Step 5: *Calculation of neural network output (a)*: In this step, the output of the neural network is calculated as per Equation 1 or Equation 2.

Step 6: *Calculation of error (e)*: In this step, the error is calculated as the departure of output from the specified target ($e=T-a$).

Step 7: *Check for convergence*: If, calculated error happens to be at least less than the error tolerance limit ($e \leq \varepsilon$), then the training convergence is fulfilled. This indicates that training is performed successfully within a considerable number of training cycles. In case, the convergence criteria is not met within a desired period of training cycles, the training process could be terminated prematurely and another fresh training may be performed with newer settings of *Step 4*, till successful training is achieved.

In this paper, all the above mentioned steps have been performed to arrive at the desired goal of imparting successful training. It is observed from the trial runs that, out of three possible selections against epoch setting at *Step 4*, an epoch setting of 10 yields significantly faster convergence as compared to its counterparts of 100 and 1000 epochs per training cycle, as indicated in Table 3.

The remaining simulation is carried out with the epoch setting of 10 and the training performance showing either full convergence or partial convergence in respective groups are presented in Table 4. The authors have deeply analyzed the results presented in Table 4 and from the analysis it is inferred that these results demonstrate some of the following interesting findings.

(i) The training performance of five qualifying cases, out of nine cases of Table 2 having single hidden layer (HL=1), is presented in the upper section of Table 4. In this category, full convergence is guaranteed for three cases (TC values 11, 12, and 13, corresponding to all the six modes of backpropagation) and partial convergence is recorded for two cases only (TC values 21, and 31, with four modes of backpropagation namely GD, GDM, GDA, and GDX). The remaining four cases (TC values 22, 23, 32, and 33), not figuring in Table 4, did not converge at all, for any of the backpropagation schemes, in as much as 999 training cycles.

(ii) The training performance of fifteen qualifying cases, out of twenty seven cases of Table 2 having two hidden layers (HL=2), is presented in the lower section of Table 4. In this category, full convergence is guaranteed for eight cases (TC values 111, 112, 113, 211, 212, 213, 312, and 313, corresponding to all the six modes of backpropagation) and partial convergence is recorded for seven cases only (TC values 121, 131, 221, 231, 311, 321, and 331, with four modes of backpropagation namely GD, GDM, GDA, and GDX). The remaining twelve cases (TC values 122, 123, 132, 133, 222, 223, 232, 233, 322, 323, 332, and 333), not figuring in Table 4, did not converge at all, for any of the backpropagation schemes, in as much as 999 training cycles.

(iii) The performance results of Table 4 also indicate successful training supported with guaranteed and faster convergence for the proposed historical Elman neural network for both the cases of hidden layer combinations (HL=1 and HL=2) having any number of neurons in these layers (in the range 1 through 5), subject to fulfillment of any/all of the conditions indicated below.

- The transfer function assigned to the output layer is purely linear.
- The transfer function assigned to the hidden layer preceding the output layer is purely linear.
- The transfer function assigned to the output layer and/or the preceding hidden layer is purely linear.

(iv) In some cases, where the transfer functions assigned (either for the output layer or the preceding hidden layer or both of them) were other than purely linear functions (i.e. tan-sigmoid or log-sigmoid), the training process did not show proper convergence. The results include either no convergence or delayed convergence.

(v) The faster and almost guaranteed convergence results shown in Table 4 are mostly dominated by neural network structures having 4, or 5 neurons in the hidden layers (NL=4, NL=5). The overall performance ratio could be evaluated from the estimate of ***FST*** ratio (i.e. the ratio of counts yielding ***F***aster performance to that of ***S***uccessful performance out of the ***T***otal cases tried). A numerical count of these terms in view of the findings of Table 4 indicates that, ***F***:***S***:***T*** = 10 : 20 : 27 ≈ 1 : 2 : 3, which is convincing and acceptable for the application of neural network based analysis.

The best part of the observation from the results of Table 4 has been recorded against TC value of 212 having two hidden layers and four neurons per layer. This case gives the fastest and certainly guaranteed convergence out of twenty cases of Table 4 in a record number of training cycles (CY=1) and over a record time period (TI=1 sec) in contrast to the computational time of conventional approach (t=946 sec). It is therefore claimed that the proposed Elman neural network approach is around 1000 times faster than the conventional approach. The graphical results supporting output input correspondence (OIC), error surface (ES), training performance (TRP), training states (TRS), and receiver operating characteristic (ROC) as obtained against the best training parameter assignment of Table 4 (TC=212, HL=2, NL=4), are presented in Fig.7 through Fig.11.

Figure 7 shows the output~input characteristic (OIC) during whole of the training process for the best training parameter assignment of Table 4 (TC=212, HL=2, NL=4). The information displayed in this figure portrays that the proposed neural network is capable of generating a valid non-zero output corresponding against all of the input dataset presented to the proposed neural network. Therefore, this observation validates appropriate selection of the neural network and its training performance.

The error surface (ES) plot of Fig.8 corresponding to the best training parameter assignment of Table 4 (TC=212, HL=2, NL=4), shows that, there exists a point on the error surface plane or the error contour plane, where least error is observed. At this point (marked by the white spot), the weights and bias parameters have reportedly the best values during the training.

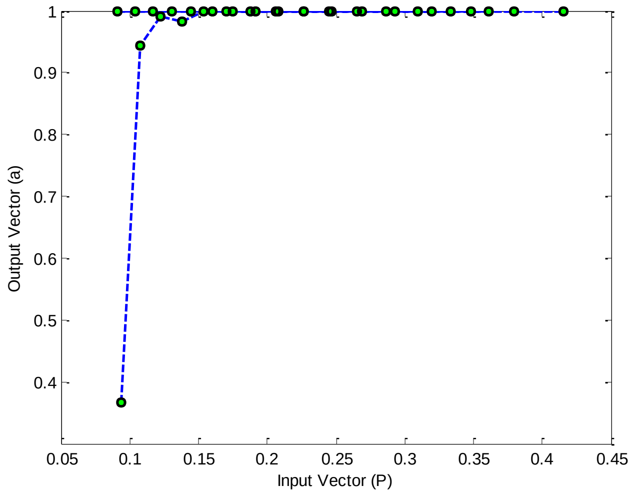


Fig. 7. Output~Input plot for case (TC=212, Table 4)

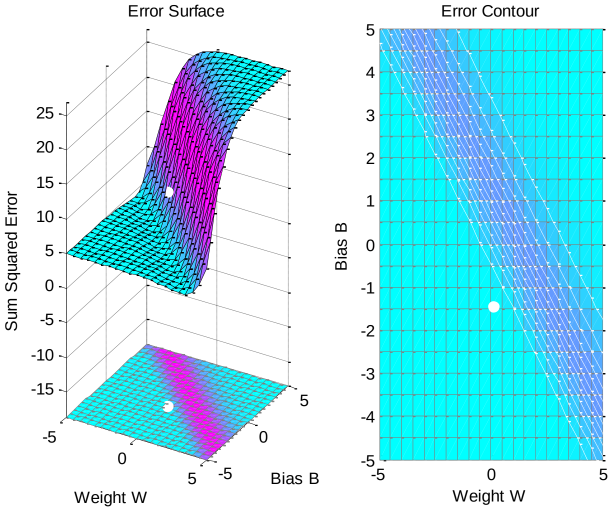


Fig. 8. Error Surface plot for case (TC=212, Table 4)

Figure 9 indicates the training performance (TRP) corresponding to the best training parameter assignment of Table 4 (TC=212, HL=2, NL=4). This plot demonstrates the variation of error during the training epochs and its relative distance from the best performance as set by the training goal. It is also observed from the plot that the error remains steady during throughout the epochs during that training cycle, which ensures stable convergence criteria for the proposed Elman neural network.

Figure 10 highlights the training states (TRS) corresponding to the best training parameter assignment of Table 4 (TC=212, HL=2, NL=4). The information contained in this plot includes error gradient and learning rate corresponding to the same training cycle.

Figure 11 indicates the receiver operating characteristic (ROC) corresponding to the best training parameter assignment of Table 4 (TC=212, HL=2, NL=4). It justifies perfect training criterion as the curve is stretched towards the upper left corner of the plotting frame. Hence, the proposed scheme meets the desired training perfection.

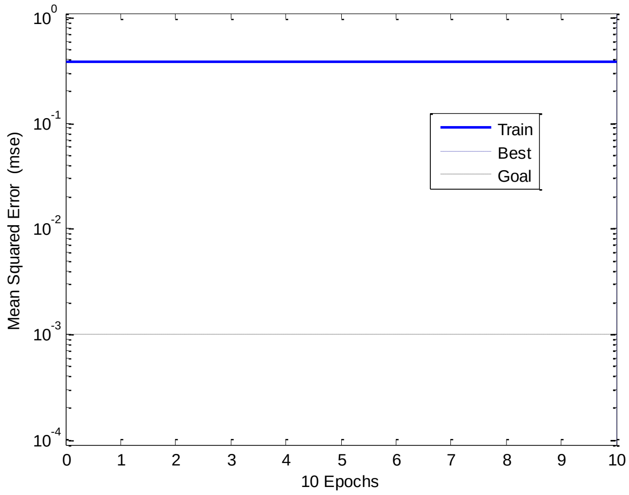


Fig. 9. Training Performance for case (TC=212, Table 4)

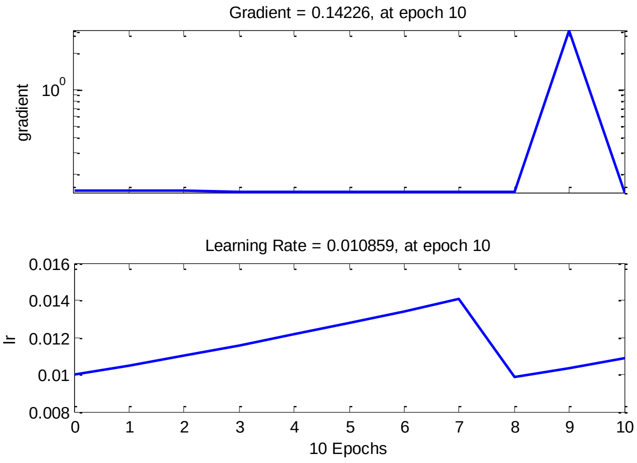


Fig. 10. Training States for case (TC=212, Table 4)

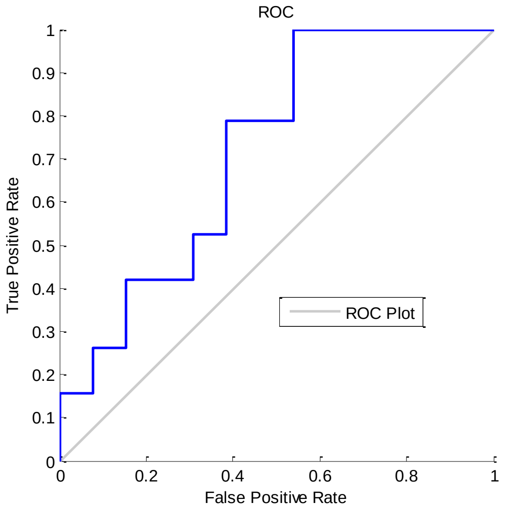


Fig. 11. ROC plot for case (TC=212, Table 4)

6. Conclusions

This paper considers successful application of historical Elman neural network to power systems in order to monitor critical busbars. This objective is fulfilled in this paper considering various aspects such as selection of input data, target data, network architecture, weights and bias parameter so as to obtain successful training based on supervised learning. The training of the neural network for achieving successful training is also validated in the results through the case study conducted on IEEE 30-bus test system, that ensures the justifications and guarantees sure and faster training convergence for the proposed neural network. It is also claimed through the findings that the Elman neural network approach with backpropagation scheme is around 1000 times faster than the conventional approach. It is expected that this outcome may be best exploited by utilities and researchers in power sector to overcome the difficulties posed by conventional approach. The utilities are expected to benefit a lot from this approach by saving time and money without sacrificing much on accuracy.

References

[1] C. Concordia, “Voltage Instability”, *Electrical Power and Energy Systems*. Vol. 13, pp. 14-20, 1991.

[2] N. Yorino, H. Sasaki, Y. Masuda, Y. Tamura, M. Kitagawa, and A. Oshimo, “An investigation of Voltage Instability Problems”, *IEEE Trans on Power Systems*. Vol. 7, pp. 600-611, 1992.

[3] V. Ajjarapu and B. Lee, “Bibliography on Voltage Stability”, *IEEE Trans on Power Systems*. Vol. 13, pp. 115-125, 1998.

[4] P. Kessel and H. Glavitsch, “Estimating the Voltage Stability of a Power System”, *IEEE Trans on Power Delivery*. Vol. 1, pp. 346-354, 1986.

[5] M.E. Hawary, *Electric Power Applications of Fuzzy Systems*, IEEE Press, 1998.

[6] L. Zadeh, “Fuzzy Sets as a basis for theory of Possibility”. *Fuzzy Sets and Systems*. Vol. 1, pp. 3-28, 1978.

[7] H.J. Zimmerman, *Fuzzy Set Theory and its Application*, Kluwer Academic Press, 1994.

[8] A.A.E. Keib and X. Ma, “Application of Artificial Neural Networks in Voltage Stability Assessment”, *IEEE Trans on Power Systems*. Vol. 10, pp. 1890-1896, 1995.

[9] T.S. Dillon, “Artificial neural nets applications to power systems and their relationships to symbolic methods”, *Electrical Power and Energy Systems*. Vol. 13, pp. 66-72, 1991.

[10] S.N. Pandey, S. Tapaswi, and L. Srivastava “Price Prediction based Congestion Management using growing RBF Neural Network”, *Proc. IEEE India Conference (INDICON)*, pp. 482-487, 2008.

[11] B. Alberto, D. Maurizio, M. Marco, S.P. Marco, and M. Politecnico, “Congestion Management in a Zonal Market by a Neural Network Approach”, *European Transactions on Electric Power*. Vol. 19, pp. 569-584, 2009.

[12] S.S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, India, 1999.

[13] M. Hagan, H. Demuth, and M. Beale, *Neural Network Design*, PWS Publishing, Boston, 1996.

[14] Z. Zhigang and W. Jun, *Advances in Neural Network Research and application*, Springer-Verlag, Berlin, 2010.

[15] P.K. Kalra, A. Srivastava, and D.K. Chaturvedi, “Artificial neural nets applications to power systems operation and control”, *Electric Power Systems Research*. Vol. 25, pp. 83-90, 1992.

[16] S. Iman, K. Abbas, and F. Rene, “Radial Basis Function Neural Network Application to Power System Restoration Studies”, *Computational Intelligence and Neuroscience*. Vol. 1, pp. 1-10, 2012.

[17] A.R. Bahamanyar and A. Karami, “Power System Voltage Stability Monitoring using Artificial Neural Networks with a Reduced set of Inputs”, *Electrical Power and Energy Systems*. Vol. 58, pp. 246-256, 2014.

[18] D.Q. Zhou, U.D. Annakkage, and A.D. Rajapakse, “Online Monitoring of Voltage Stability Margin Using an Artificial Neural Network”, *IEEE Trans on Power Systems*. Vol. 25, pp. 1566-1574, 2010.

[19] M. Mirzaei, M.Z.A. Kadir, H. Hizam, and E. Moazami, “Comparative Analysis of Probabilistic Neural Network, Radial Basis Function, and Feed-forward Neural Network for Fault Classification in Power Distribution Systems”, *Electric Power Components and Systems*. Vol. 39, pp. 1858-1871, 2011.

[20] L. Xue, C. Jia, and D. Dajun, “Comparison of Levenberg-Marquardt method and Path Following interior point method for the solution of optimal power flow problem”, *Emerging Electric Power Systems*. Vol. 13, pp. 15-35, 2012.

[21] B. Ilamathi, V.G. Selladurai, and K. Balamurugan, “ANN-SQP Approach for NOx emission reduction in Coal Fired Boilers”, *Emerging Electric Power Systems*. Vol. 13, pp. 1-14, 2012.

[22] D.K. Ranaweera, “Comparison of neural network models for fault diagnosis of power systems”, *Electric Power Systems Research*. Vol. 29, pp. 99-104, 1994.

[23] El. Sharkawi, M.A. Marks, and R.J. Weerasoritya, “Neural Networks and their Application to Power Engineering”, *Control Dynamic System, Advances in theory and Applications*. Vol. 41, 1991.

[24] M.F. Moller, “A scaled conjugate gradient algorithm for fast supervised learning”, *IEEE Trans. on Neural Networks*. Vol. 6, pp. 525-533, 1993.