Available online at www.sciencedirect.com
ScienceDirect
Electrical Power and Energy Systems 30 (2008) 60–66
ELECTRICAL POWER & ENERGY SYSTEMS
www.elsevier.com/locate/ijepes

Radial basis function neural network for power system load-flow

A. Karami *, M.S. Mohammadi

Faculty of Engineering, The University of Guilan, P.O. Box 41635-3756, Rasht, Iran

Abstract

This paper presents a method for solving the load-flow problem of the electric power systems using radial basis function (RBF) neural network with a fast hybrid training method. The main idea is that some operating conditions (values) are needed to solve the set of non-linear algebraic equations of load-flow by employing an iterative numerical technique. Therefore, we may view the outputs of a load-flow program as functions of the operating conditions. Indeed, we are faced with a function approximation problem and this can be done by an RBF neural network. The proposed approach has been successfully applied to the 10-machine and 39-bus New England test system. In addition, this method has been compared with that of a multi-layer perceptron (MLP) neural network model. The simulation results show that the RBF neural network is a simpler method to implement and requires less training time to converge than the MLP neural network.

Keywords: Power systems load-flow; Radial basis function neural network; Multi-layer perceptron neural network

1. Introduction

The load-flow (power-flow) problem is one of the most important tasks performing by operation engineers of power systems and its main objective is calculation of all bus voltages magnitudes and angles, and consequently the power flow over the transmission lines [1,2]. In fact, the steady state operating conditions of the power systems are determined by performing a load-flow analysis on the underlying systems.

Continuous growth and complexity of the power systems have originated the adoption of sophisticated computer methods for efficient planning, operation and control of their systems [3]. Usually, these computer methods perform the load-flow analysis on the systems. The load-flow analysis is intrinsically a time-consuming task, because the set of non-linear algebraic equations of load-flow are, in general, solved by employing iterative numerical methods. Therefore, these numerical methods are not fully suitable for on-line applications.

In recent years, the neural networks (NNs) have been proposed for solving many different problems of the power systems such as the static and dynamic security assessments; e.g., see [4–9], as they have the ability to provide the value of the severity indices accurately. The neural networks inputs developed in these references are all determined through a load-flow analysis on the systems. However, as mentioned above, the load-flow problem is a time-consuming task especially in large power systems, but there has been made less attention in applying NNs to the load-flow problem. For example, to perform on-line static security assessment, the computation of many load-flow scenarios in a few minutes are needed [5].

In reference [10], a complicated architecture of the dynamic neural networks has been developed to implement the Newton–Raphson algorithm for solving the set of non-linear equations of load-flow analysis. In our previous work [11], we demonstrated that the multi-layer perceptron (MLP) neural networks could be used for the estimation of the outputs of a load-flow program (or simply "load-flow outputs") in a small test system having three machines

* Corresponding author. Tel.: +98 131 6690270; fax: +98 131 6690271.
E-mail addresses: karami_al@yahoo.com, karami_s@guilan.ac.ir (A. Karami).

0142-0615/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijepes.2007.10.004

and nine buses. The main idea is that some operating conditions are needed for solving the set of non-linear algebraic equations of load-flow by employing an iterative numerical technique. Therefore, we may view the outputs of a load-flow program as functions of the operating conditions and the MLP NNs may be employed to approximate these functions. The same idea was also applied in Ref. [3]. However, in both references, a separate MLP NN has been used for the estimation of each load-flow outputs. Thus, we need to train a large number of NNs in a given power system and this idea is impractical.

Furthermore, in Ref. [3] some indirect system variables have been chosen as the inputs of NNs. Therefore, determination of these inputs is itself a time-consuming task, because for determination of these inputs supplementary software is required. Therefore, the high-speed solution capability of a neural network has not been fully exploited in this reference. In addition, it would be better to use "raw" system data as the inputs of the NNs. Because, the correct system operating conditions may not be given to the NNs by using indirect system variables as the inputs. The reason is that different raw system data may lead to the same indirect system variables.

In this paper as in [11], we use direct or raw system measurements as the inputs of NNs and apply our method on the 10-machine and 39-bus New England test system. Here, only one RBF neural network with a fast hybrid training method is employed for the estimation of all load-flow outputs in the test system almost instantaneously. In addition, the proposed method is compared with an MLP NN model and the results are given.

2. Radial basis function (RBF) neural network

The basic structure of the RBF NN used in the present paper is shown in Fig. 1 [12]. The RBF NN consists of three layers namely, the inputs layer, the hidden (or RBF) layer, and the output layer. The nodes within each layer are fully connected to the previous layer. The input nodes are directly connected to the hidden layer neurons. The output of the jth hidden neuron can be written as [12]

$$h_j = \exp\left[-\frac{\|\mathbf{X} - c_j\|^2}{\sigma_j^2}\right] \tag{1}$$

where h_j is the output of the jth neuron, $\mathbf{X}$ is the input vector, c_j is the neuron's center and σ_j is the center spread parameter. Here, we have used Gaussian transfer function for the hidden neurons. The neurons of the output layer have a linear transfer function. It is simply the weighted summation of the outputs of all hidden neurons connected to that output neuron. For the kth neuron the output y_k is [12]

$$y_k = \sum_{j=1}^{m} W_{kj} \cdot h_j \tag{2}$$

where W_{kj} is the synaptic weight connecting hidden neuron j to output neuron k and m is the number of the hidden layer neurons.

Training RBF NN is aimed at adjusting Gaussian basis function centers (c_j's) and spread parameters (σ_j's), and weights (W_{kj}'s) to result in minimum sum-squared error for all the output units among all the patterns [13,14]. In this paper as in [15], first, the number of hidden neurons and the neuron centers are determined by some form of the clustering algorithm. Then, the spreads are determined by a P-nearest neighbor method [16]. Finally, the weights between the hidden and the output layers are computed using the pseudo-inverse technique. Euclidean distance-based clustering method [17] has been employed in this paper to select the number of hidden neurons and the neuron centers. The normalized input and the output data are used for training of the RBF NN.

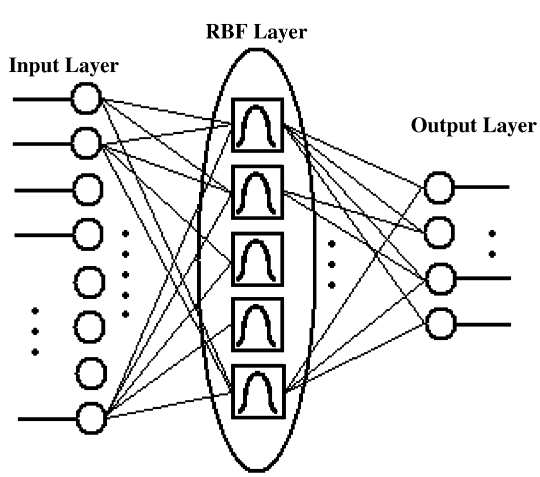

Fig. 1. Radial basis function (RBF) neural network structure.

3. The proposed RBF neural network methodology

In this section, we present our RBF NN based approach by applying it to the 10-machine and 39-bus New England test system shown in Fig. 2. The system data are given in [18]. Although the method is being applied to this test system only, it is quite a general one and can be used for any other system in a similar manner.

As mentioned before, the outputs of a load-flow program are functions of the system operating conditions. Therefore, we may use an RBF neural network to approximate the functions. The inputs of this neural network are being the system operating conditions and its outputs are being the load-flow outputs. The input variables for the proposed neural network need to be carefully selected, in attempting to emulate the solution process of a conventional load-flow computer program. Thus, we try to extract a proper set of system operating conditions in the New England test system as follows.

We assume that bus 1 represents the slack bus whose voltage magnitude and voltage angle are known. The remaining generation buses (i.e., buses 2–10) are considered

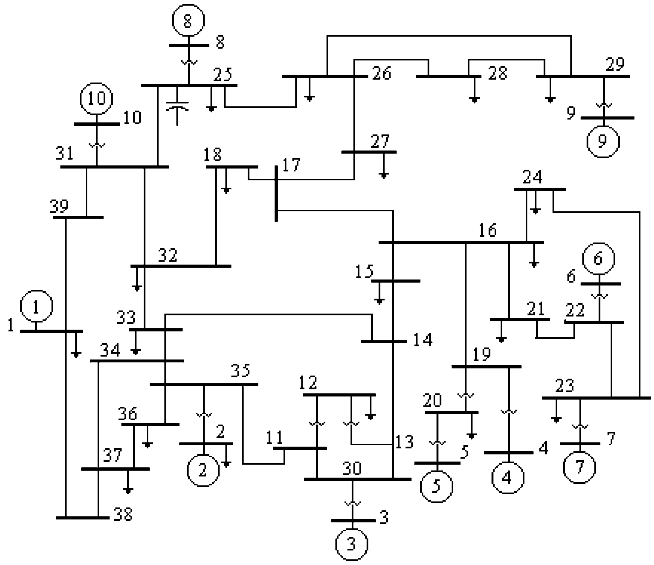

Fig. 2. One-line diagram of the New England test system.

as PV buses whose generated active powers and voltage magnitudes are denoted by PG_i and V_i ($i = 2$–10). Besides one slack bus and 9 PV buses, the test system consists of an additional 29 PQ buses (i.e., buses 11–39). However, the loads are acting only on 19 distinct buses. The active and reactive load powers of these buses are denoted by PD_j and QD_j (j is the bus number). Note that the shunt capacitor installed at bus 25 is treated as a load whose generated reactive power is known as Q_c.

Without loss of generality, we assume that the voltage magnitude and the voltage angle of the slack bus (i.e., bus 1) are fixed at their assumed specified values. Therefore, all the operating conditions in the New England test system are

– voltage magnitudes of all 9 PV buses (V_2–V_{10}).
– Generated active powers of all 9 PV buses (PG_2–PG_{10}).
– Active load powers of all 19 loads acting on different buses ($PD_1, PD_2, \ldots, PD_{37}$).
– Reactive load powers of all 19 loads acting on different buses ($QD_1, QD_2, \ldots, QD_{37}$).
– Generated reactive power of the single shunt capacitor installed at bus 25 (Q_c).

Hence, all the load-flow outputs are functions of the above $(9 + 9 + 19 + 19 + 1) = 57$ operating conditions. To approximate this function with an RBF neural network is the main objective of this paper. Here, we want to estimate all the load-flow outputs by using only one RBF NN. This NN is shown in Fig. 3, in which the load-flow outputs have been represented by Output 1, Output 2, ..., Output n. The outputs are, respectively, active and reactive generated powers of the slack bus, the reactive generated powers of 9 PV buses, the voltage magnitudes of 29 PQ buses (i.e., buses 11–39), and the voltage angles of 9 PV and 29 PQ buses (i.e., buses 2–39). Thus, the total number

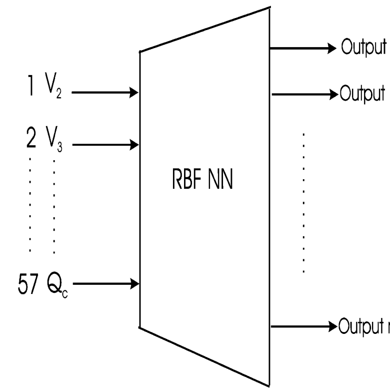

Fig. 3. Selected architecture for the RBF NN based method, $n = 78$.

of the outputs in the test system is $n = (1 + 1 + 9 + 29 + 38) = 78$.

The proposed RBF neural network needs to be trained on a limited set of cases covering the operating conditions for the test system. Once the training of the network is competed, all the load-flow outputs can be quickly computed. Because, all the 57 operating conditions are, in general, directly monitored in an energy control center (ECC).

The procedure for obtaining the training or testing data patterns is composed of the following steps:

(1) The minimum and maximum limits of the generated reactive powers in all the PV buses are set to −0.2 and 0.7 times their nominal generated active powers, respectively.
(2) It is assumed that the following operating conditions vary independently over some specified ranges:
  (a) voltage magnitudes of all 9 PV buses,
  (b) generated active powers of all 9 PV buses,
  (c) active and reactive powers of all 19 loads,
  (d) generated reactive power of the shunt capacitor located at bus 25.

  It is further assumed that the range of variations of the voltage magnitudes of PV buses is bounded from 0.9 to 1.1 times their corresponding nominal values. The range of variations of the other operating conditions is bounded from 0.6 to 1.1 times their corresponding nominal values.
(3) A random value with uniform distribution is assigned independently to each of the variables mentioned in step 2.
(4) With the above-prepared data, an AC load-flow is performed. The results may show that some PV buses may be treated as PQ buses. However, the voltage magnitudes of all the violated and non-violated PV buses are used as the inputs of the proposed neural network.

Here we have assumed that the reactive power generation limits of the PV buses are fixed for all operating conditions, and therefore these limits have not been chosen as the inputs of the proposed NN. If these limits are changed, we can use them as the additional inputs of the NN.

It should be noted that the unsolvable load-flow training data will not be considered as the training patterns. However, the trained NN will produce outputs for even unsolvable load-flow problems. To exclude these load-flow outputs, we can check the outputs to see if the large violations are occurred. For example, if the voltage magnitudes of some buses being too low or too high, we find that the current operating point is unsolvable, and it is not a real operating point.

4. Numerical results and discussion

4.1. Simulation results

Simulation tests have been carried out on the New England test system. With the procedure presented in Section 3, a database of 4000 operating conditions was built by an AC load-flow program from which 3000 cases were chosen for training and the remaining 1000 cases for testing of the proposed RBF NN (see Fig. 3). Before training, the input and output data patterns were scaled so that they fell in the range [0.1–0.9].

As mentioned before, we have used a hybrid (unsupervised/supervised) method for training the proposed RBF NN. In unsupervised training phase of this hybrid method, the input patterns are clustered according to the similarities discovered among the input features. The clustering process is governed by a threshold called the "vigilance" parameter and Euclidian metric function. In the clustering, the first pattern is selected as the center of the first cluster. Then, the next pattern is compared with that of the first cluster center. If the distance is less than the vigilance parameter, it is clustered with the first. Otherwise, it is a center of a new cluster. This process is repeated for all patterns. Once all patterns are processed, the algorithm is reiterated until a stable cluster formation occurs. This method was firstly proposed in [19] and then used in [15] for RBF NNs. In supervised phase of training method, the weights between the hidden and output layers are computed directly using the pseudo-inverse technique.

By employing the above-mentioned clustering method with different values for vigilance parameter, we came up with different number of hidden neurons for the RBF NN. In each case, after finding the structure of the RBF NN and its parameters, the trained NN was tested using 1000 test data patterns. The performance of the trained NN was examined by the mean-squared error (MSE) between the actual and the estimated outputs.

The MSE values are shown in Fig. 4 against the number of hidden layer neurons. Regarding this figure and by considering both of the structure of the RBF NN and the value of MSE for test data patterns, we chose 993 hidden neurons as the sub-optimal number of the hidden neurons for the proposed RBF NN. For this choice, the obtained value of MSE for 1000 test patterns was 0.0014. In unsupervised training phase of the network, 16 iterations were performed to form stable cluster with vigilance parameter equal to 1.77. Total elapsed time for training the RBF NN was only 4 min on a Pentium IV 1.6 GHz personal computer.

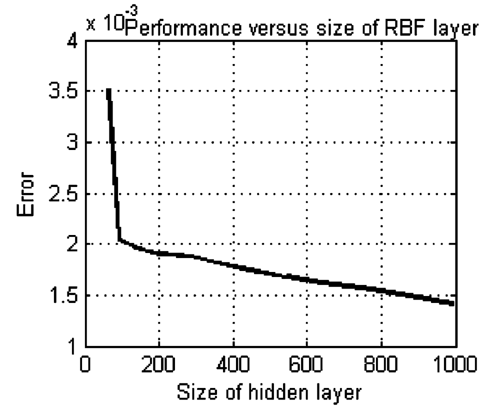

Fig. 4. The mean-squared error (MSE) of test pattern versus number of hidden neurons of the RBF NN.

Fig. 5 shows the MSE value of test pattern corresponding to 78 outputs. As can be seen from this figure, Output 2 (i.e., QG_1) has the biggest MSE and Output 21 (i.e., V_{20}) has the least MSE. Since different load-flow outputs show variation in different ranges, we have obtained different values for the MSE of the outputs. From Fig. 5 we see that the performance of the trained RBF NN is quite good. To see this better, Fig. 6 compares the actual and estimated corresponding values of the 78 outputs of the RBF NN for the best and worst results among 1000 test patterns. As can be seen from this figure, the trained RBF NN can estimate the actual outputs very well almost instantaneously. Therefore, the proposed method is well suitable for on-line load-flow analysis.

In order to compare the proposed RBF NN based approach with an MLP NN model, we chose 1000 training patterns among 3000 training patterns to train an MLP NN. This NN has 57 inputs and 78 outputs, as the same as those used for RBF NN. An MLP neural network generally consists of an input layer, an output layer, and one

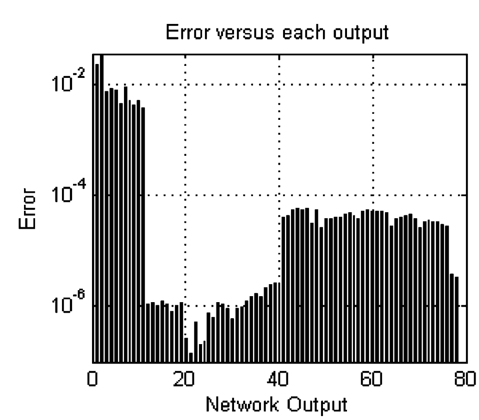

Fig. 5. The mean-squared error (MSE) values for 78 outputs of the RBF NN.

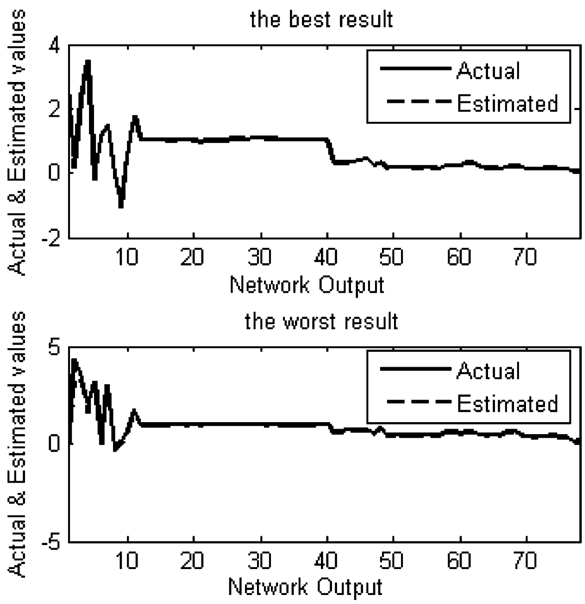

Fig. 6. Comparison of the actual and estimated corresponding values of the 78 outputs of the RBF NN.

or more hidden layers with each layer having a number of neurons.

One of the popular training algorithms of the MLP NNs is the error back propagation algorithm [13], which is based on the gradient descent technique for error reduction. However, the standard back propagation method is too slow for many applications. In this paper, we used the MATLAB neural network toolbox [20] to train the MLP NN, and *Resilient* back propagation technique was employed for this purpose. This is a fast training method for the MLP NNs.

For the MLP NNs, the number of hidden layer and hidden layer neurons are obtained by using a trial-and-error approach. Therefore, an MLP NN with one hidden layer was tried first, but was found hard to converge. Thus, an MLP NN with two hidden layers was selected for further analysis. Sigmoid transfer functions were used for the hidden layers neurons and linear transfer functions were used for the output layer neurons. After running different simulations, we chose twenty hidden neurons for both hidden layers of the proposed MLP NN. Before training, the input and output data patterns were scaled so that they fell in the range $[-1, 1]$. The error goal, the MSE between actual and estimated outputs in the training phase was set to 0.0005. More that 50,000 training iterations were performed to converge and it took more than 10 h to reach this value for the MSE.

In order to see how well the trained MLP NN generalizes, the same 1000 test patterns used for the RBF NN were given to the MLP NN. The obtained value of MSE for the test patterns was 0.0014 proving the generalization accuracy of the trained MLP NN. Therefore, the MLP NN estimates the load-flow outputs in the New England test system almost instantaneously as the trained RBF neural network does. However, the training time of the MLP NN is much more than that of the RBF NN based approach. To see this better, Fig. 7 shows the MSE values of test patterns corresponding to 78 outputs. In addition, Fig. 8 compares the actual and estimated corresponding values of the 78 outputs of the MLP NN for the best and worst results among 1000 test patterns.

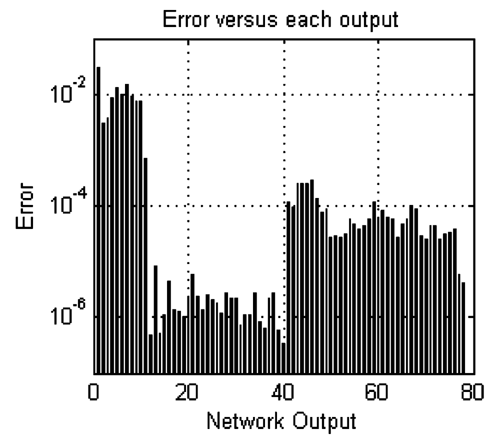

Fig. 7. The mean-squared error (MSE) values for 78 outputs of the MLP NN.

In order to further compare the performance of the trained RBF and MLP neural networks, we chose an operating point, which has been sufficiently different from the points in the training set. The Euclidean distance between this point and the closest one in the training set was 18.57. For this operating point, the mean-squared error (MSE) between the actual and estimated outputs using the trained RBF and MLP NNs were respectively obtained 0.0643 and 0.6852. The small error obtained by the trained RBF NN confirms that this network has estimated the

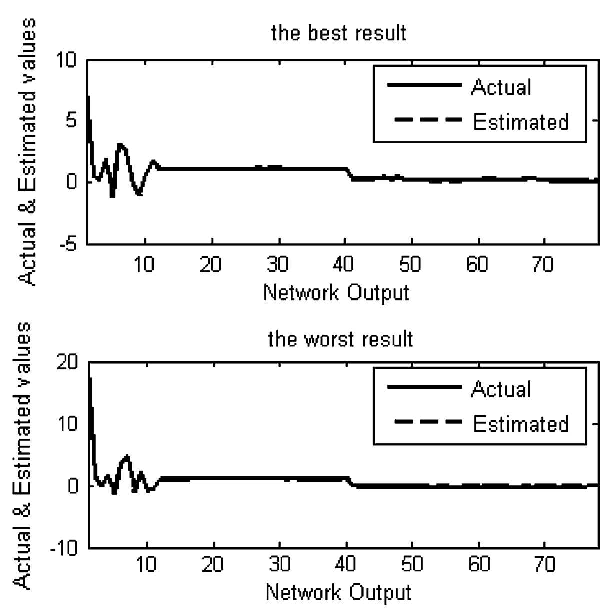

Fig. 8. Comparison of the actual and estimated corresponding values of the 78 outputs of the MLP NN.

load-flow outputs with a good degree of accuracy. However, the performance of the trained MLP NN was not as good as. To see this better, Figs. 9 and 10 compare the actual and estimated corresponding values of the 78 outputs of the RBF and MLP neural networks. As can be seen from these figures, the estimated outputs of the RBF NN are very close to the actual outputs. It should be noted that after training completion of the RBF NN, the generalization performance of this NN depends on the distance between the system operating point and the centers of the hidden (RBF) layer neurons. Thus, the above results show that the RBF NN has correctly been trained.

4.2. Discussion

As mentioned before, the load-flow outputs are functions of the system operating conditions. So, the system

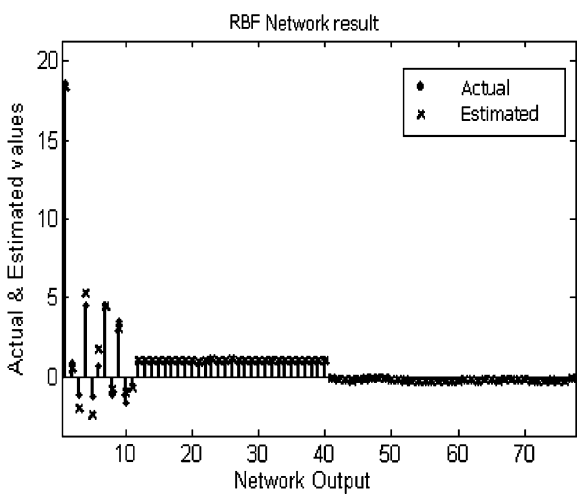

Fig. 9. Comparison of the actual and estimated corresponding values of the 78 outputs of the RBF NN for an operating point far from training patterns.

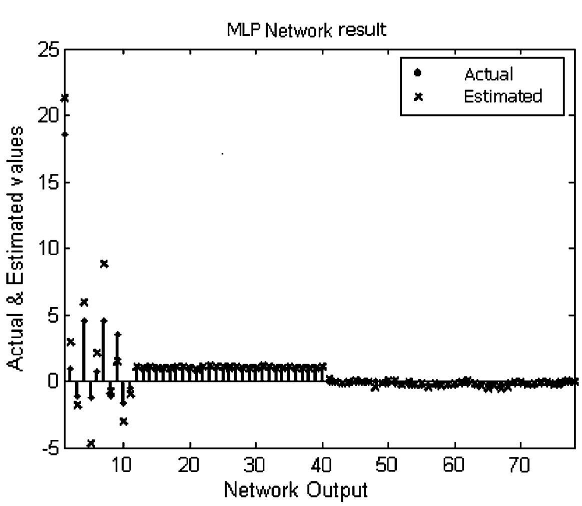

Fig. 10. Comparison of the actual and estimated corresponding values of the 78 outputs of the MLP NN for an operating point far from training patterns.

operating conditions were chosen as the inputs of the proposed RBF NN (see Fig. 3). It is obvious that if any of the system operating conditions changes, we should expect to have some changes on the load-flow outputs. In a much larger power system, the inputs of the proposed NN get larger. If a large number of inputs are used, the number of interconnection weights and the required training data vectors will increase. To cope with this problem, some feature selection technique [17,21–23] may be used to select those inputs (features) that contribute the most to the discrimination ability of the NN. Only these features are then used to train the NN and the rest are discarded.

Another method for the proposed NN inputs reduction is decomposing of the power system into areas, in which different NNs will perform the load-flow analysis. However, the simulation results presented in this paper show that for a medium-sized power systems, such as the New England test system, only one RBF NN can be employed for load-flow analysis.

The method proposed in this paper suggests the use of an RBF NN as a load-flow solver of a specific power system. Therefore, only the operating conditions such as the voltages magnitudes and the generated active powers of PV buses, the active and reactive powers of loads and the generated reactive power of the shunt capacitors were chosen as the inputs of the NNs. However, the proposed neural network can also be trained using other operating data such as the transformer tap ratios, the phase shifter angles, and the shunt reactor compensations.

We know that the power system topology is being changed due to transmission lines or generators outages. In these cases, the trained NN for a specific topology will fail to estimate the accurate results for the load-flow outputs as it would be unable to capture the inputs-outputs relationship properly. A solution for this problem is to train a separate RBF NN for each possible system topology. Another solution is to use the transmission lines or generators status as the additional inputs for the proposed RBF NN. Using the above aspects in a much larger power system, therefore, present directions for further research.

5. Conclusions

A radial basis function (RBF) neural network (NN) based approach has been presented for power system load-flow analysis. The inputs of the proposed NN are a set of directly monitorable variables of the system. The high performance of this approach was assessed on the New England test system. Simulation results indicated that the trained NN could be used to estimate the load-flow outputs under different operating conditions with a high degree of accuracy almost instantaneously. A fast hybrid method was used for training the RBF NN. First, Euclidean distance-based clustering technique was employed to select the number of hidden neurons and neuron centers. Then, weights between the hidden and output layers were computed directly using the pseudo-inverse technique.

In addition, the RBF NN based method was compared with that a multi-layer perceptron (MLP) NN model. The MLP NN was also able to estimate the load-flow outputs in the test system. However, the training time of the MLP NN was much more than that of the RBF NN. The results also show that when the operating point of the power system is sufficiently different from those in the training set, the RBF NN produces the better estimations for the load-flow outputs than the MLP NN does. The general conclusion, therefore, is that the RBF NN should be preferred to MLP NN for on-line load-flow analysis in energy control centers (ECC).

References

[1] Stevenson WD. Elements of power system analysis. 4th ed. McGraw-Hill; 1982.
[2] Glover JD, Sarma M. Power system analysis and design with personal computer applications. 2nd ed. Boston: PWS Publishing Company; 1994.
[3] Paucar VL, Rider MJ. Artificial neural networks for solving the power flow problem in electric power systems. Electr Power Syst Res 2002;62:139–44.
[4] Sobajic DJ, Pao YH. Artificial neural-net based dynamic security assessment for electric power systems. IEEE Trans Power Syst 1989;4(1):220–8.
[5] Weerasooriya S, El-Sharkawi MA, Damborg M, Marks II RJ. Towards static security assessment of a large scale power system using neural networks. IEE Proc—C 1992;139:64–70.
[6] Zhou Q, Davidson J, Fouad AA. Application of artificial neural networks in power system security and vulnerability assessment. IEEE Trans Power Syst 1994;9(1):525–32.
[7] Hobsen E, Allen GN. Effectiveness of artificial neural networks for first swing stability determination of practical power systems. IEEE Trans Power Syst 1994;9(2):1062–8.
[8] Aboytes F, Ramirez R. Transient stability assessment in longitudinal power systems using artificial neural networks. IEEE Trans Power Syst 1996;11(4):2003–10.
[9] Mansour Y, Chang AY, Tamby J, Vahedi E, Corns BR, Sharkawi MA. Large scale dynamic security screening and ranking using neural networks. IEEE Trans Power Syst 1997;12(2):954–60.
[10] Nguyen TT. Neural network load-flow. IEE Proc Gener Transm D 1995;142(1):51–8.
[11] Menhaj MB, Karami A. Neural networks for power system load-flow. In: Proceedings of the IASTED international conference on applied modelling and simulation, Banff, Canada; 1997. p. 216–20.
[12] Refaee JA, Mohandes M, Maghrabi H. Radial basis function networks for contingency analysis of bulk power systems. IEEE T Power Syst 1999;14(2):772–8.
[13] Haykin S. Neural networks: a comprehensive foundation. 2nd ed. Englewood Cliffs, NJ: Prentice-Hall; 1999.
[14] Bahbah AG, Girgis AA. New method for generators' angle and angular velocities prediction for transient stability assessment of multimachine power systems using recurrent artificial neural network. IEEE Trans Power Syst 2004;19(2):1015–22.
[15] Jain T, Srivastava L, Singh SN. Fast voltage contingency screening using radial basis function neural network. IEEE Trans Power Syst 2003;18(4):1359–66.
[16] Karayiannis NB, Mi GW. Growing radial basis neural networks: merging supervised and unsupervised learning with network growth techniques. IEEE Trans Neural Networks 1997;8(6):1492–506.
[17] Srivastava L, Singh SN, Sharma J. Comparison of feature selection techniques for ANN based voltage estimation. Electr Power Syst Res 2000;53:187–95.
[18] Pai MA. Energy function analysis for power system stability. Boston: Kluwer Academic Publishers; 1989.
[19] Pao YH, Sobajic DJ. Combined use of unsupervised and supervised learning for dynamic security assessment. IEEE Trans Power Syst 1992;7(2):878–84.
[20] Neural networks toolbox user's guide for use with Matlab 7. The Mathworks Inc., 2005.
[21] Jensen CA, El-Sharkawi MA, Marks II RJ. Power system security assessment using neural networks: feature selection using Fisher discrimination. IEEE Trans Power Syst 2001;16(4):757–63.
[22] Kwak N, Choi C-H. Input feature selection for classification problems. IEEE Trans Neural Networks 2002;13(1):143–59.
[23] Niazi KR, Arora CM, Surana SL. Power system security evaluation using ANN: feature selection using divergence. Electr Power Syst Res 2004;69:161–7.