Designing an Intelligent Decision Support System for Human-Centered Utility Management Automation
Part 2: Design Procedure, Experimental Results, and Case Study

ALIREZA FEREIDUNIAN[1,2], CARO LUCAS[2,3], HAMID LESANI[2], MANSOOREH ZANGIABADI[2]

(1) Islamic Azad University, Islamshahr Branch, Tehran
(2)Electrical and Computer Engineering Department, Faculty of Engineering, University of Tehran, Tehran
(3) Intelligent Systems Group, Essential Sciences Research Center, IPM, Tehran
IRAN

Abstract: - In this paper, a DSS design procedure is presented. The mentioned DSS is a neural network, which is used to estimate the state of a power distribution system loading condition. The effects of different sorts of data distributions, pre-processing, complex conformal mapping, input noise, and error function on the learning and recalling performance of the DSS neural networks are studied. A practical example illustrates how the finally designed DSS can aid decision and control operations in a real standard distribution system, in both normal and abnormal conditions. A mathematical discussion, on a contradiction concerning to the effect of space equalization on DSS learning is brought in an appendix.. The paper includes discussions, practical numerical examples, and results.

Keywords—Decision Support, Neural Networks, Utility Management Automation, Human-Centered Systems, State Estimation, Power Distribution System, Complex Conformal Mapping, Computational Intelligence, Power System Contingency

1 Introduction

A decision support system (DSS) provides informational support, by processing the available data and signals in a utility control room database, for decision maker. Artificial intelligent systems (e.g. Artificial Neural Networks (ANNs)) are of successful signal processing solutions for estate estimation, and consequently decision support applications. The term "Artificial Intelligence (AI)" was coined by John McKarthy in 1954 [1,2]. Nowadays, improvements in intelligent signal processing techniques have led to adaptive, self-tuning, model free, robust, non-linear, and stochastic systems. These systems have small and tolerable internal errors [3,4].

In the present work a decision support system has been introduced to aid power system human operator, using artificial neural networks. A considerable effort has been focused on testing, comparing, and proving the abilities of two well-known ANNs: MLPs and RBFs, during learning and recalling periods.

2 DSS Design Procedure

Radial Basis Functions (RBFs) and Multi Layer Perceptrons (MLPs) are the most famous pattern recognizer neural networks. These two types of neural networks have been used in this work for designing classifiers.

2.1 Radial Basis Function (RBF) Design

2.1.1 Experimental Results and Observations on Uniform Data Distribution

First of all, uniformly distributed data of Fig. 4, (Part 1 of this paper) is learnt to a RBF neural network. Since the uniformly distributed data have an scattered form, the RBF neural network -which gains “k-means clustering” unsupervised algorithm- puts a single neuron for every single sample in inner layer, i.e. almost 80 Gaussian neurons are required and generalization ability of network is catastrophic!

The clustering method algorithm puts a class for each colony of samples. So, an idea to resolve this problem is to increase Gaussian function bandwidth. If the Gaussian functions bandwidth, which in BRF nets determines the radius of elliptic or circular classes, is increased, the classes will be greater, and consequently more samples will be included in each class. In extreme case, it must converge to the correct number of classes. Unfortunately, implementing this idea makes the error matrix ill-conditioned or rank troubled. So, this learning algorithm does not converge.

As BRF nets classification spaces are hyper elliptic or hyper spherical, hyper-elliptication of the classification spaces may be a way to improve the BRF networks learning convergence. Thus, rectangular spaces of Fig.s 7 and 8 (Part 1) are transformed to circular spaces of Fig.s 9 and 10 (Part 1), using complex conformal mappings. Then they are learnt to neural network, the learning procedure were fairly satisfactory. (The four Fig.s 7 -10 are from Part 1 of this paper. We could not place them in the Part 2 again, according to the lack of space)

Random noise was added to the inputs as well. A random noise term with γ coefficient (for controlling the noise amplitude) was added to the input data. This noise term transfers the system from a pure deterministic system to a fairly stochastic system. It has a wonderful effect on the learning of the network. Without noise, there will be problems such as divide by zero or ill conditioning of error and coefficients matrices. Reducing noise coefficient (γ) reduces the number of required neurons in Gaussian layer, but it can not be decreased lower than a limit. The limit is the mentioned matrix ill conditioning problem. This matter has been illustrated in Fig. 1. It can be seen that the conformal mapping has significant effect on the number of required neurons reduction for learning RBF neural network.

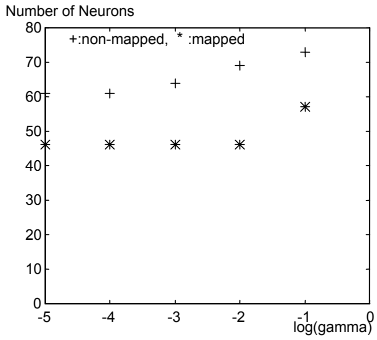

Fig. 1, Number of neurons required for network learning, versus logarithm of noise coefficient. +s show non-mapped and *s show mapped situation.

As you see in Fig. 1, applying conformal mapping reduces the number of required neurons from 60 to 45, which is a good improvement. But in this case, the weakness of generalization capability in RBF is remained. Increasing Gaussian neuron bandwidth exceeding a limit causes problems. Training error can not be reduced lower than a specific value yet, too. Besides, training methods like as Learning Vector Quantization (LVQ) and Kohonen self organizing method have tested and have not given feasible results.

2.1.2 Experimental Results and Observations on Concentrated Data Distribution

Now, let us try the concentrated data distribution. In spite of uniformly distributed data, clustered (centered) data without scaling and conformal mapping (i.e. decision space shown in Fig. 5 of Part 1 of this paper) show hopeful improvement in RBF training. These data are learned with an error less than one percent by 14 inner layer neurons. Compare 14 neurons to 45 neurons in the former case. Meanwhile, if ten percent error will be satisfactory, only two inner layer neurons are enough. Furthermore, adding input noise is not necessary.

Finally, RBF network trained by clustered (centered) data has two input, 14 Gaussian middle layer neurons and two linear output layer neurons. This is much better and smaller than the RBF network trained by uniform data, in order of dimension.

The generalization capability of this network is much higher than the network trained by uniformly distributed data, too. It can respond to training and test set inputs with an acceptable precision. The reason of this can be found in Gaussian characteristics of RBF network transfer functions. Hence, *"RBF neural networks are center oriented classifier neural network, and it must be focused on class centers for training it"*.

2.2 Multi Layer Perceptron (MLP) Design

2.2.1 Designing MLP Topology and Learning Method

"Error Back Propagation" learning algorithm is used for training Multi Layer Perception. Also, "Cross Validation" method [3,4] is used in order to test and increase the ability of learned neural network in responding to new inputs (generalization). So that, 20 percent of samples (data) are selected randomly as test set, and the residue is taken as training set. Learning will be continued, unless the both of training set error (E_{Trn}) and the test set error (E_{Tst}) decrease. The advantage of this method is preventing "over-training" or "over-fitting" of network which graphically saying is decreasing E_{Trn} curve when E_{Tst} curve is increasing.

For the reason of having two inputs, the neural network must have two input units. The DSS must produce one of the four possible pre-defined states. For designing output layer three essential topologies are suggested:

1. One linear single neuron which generates four numbers of 1, 2, 3, 4, as four pre-defined classes.
2. Two sigmoid neurons which generate four binary output numbers, as a BCD (Binary Coded Decimal).

3. Four sigmoid neurons which one of them having greatest output shows the estimated state, as one-hot output.

One advantage of topology No. 1 is having physical interpretation. The network can use hidden logic underlying in its structure for better learning. The network output for new inputs may not be integer, so decision about system state (class) will be confusing in such a cases (i.e. non-integer output). Another disadvantage of this topology is its less capability in error tolerance. For example, a 0.7 error in topology No. 1 can change the result. But, in topology No. 3 even if the output of one of wrong neurons were 0.7 (instead of zero), the correct neuron, is selected as proper activated output neuron, because it has a greater value (i.e. one).

Another subject which has been noted in neural network literature as a decision making parameter about network topology [3,4] is the output mapping space. It is apparent that the distance between the classes in a four-dimensional output space is rather greater than the distances in a one-dimensional space. This subject may be a reason for higher precision in the topology No.3. It can be said that this subject is another presentation or physical-topological interpretation for fault tolerability problem.

However, another approach to this case is that the topologies having less neuron, for the reasons of lower dimensions, less weights and smaller optimization space may have less local optimums, and consequently easier learning. This is an advantage for the first topology. Also, requiring a BCD to decimal decoder is a disadvantage for second topology.

Finally, and more important than every theoretical, philosophical, and physical interpretation, numerous practical experiments show that the third topology (the network having four outputs) has better training and generalizing performance than both the other networks. Having physical interpretation in topology No. 1 is not very important. Because, artificial neural networks have four major properties of non-linearity, learning ability, finding hidden properties of classes and being model free.

In order to make decision on the number of neurons in the hidden layer a combination of two famous methods: cascade correlation (based on network growth) and pruning (deleting weights and neurons having less contribution in generating the output) [3,4].

2.2.2 Experimental Results and Observations

It has been concluded from experimental studies, that the optimum neural network is a three layer perceptron having two nodes in the input layer, three sigmoid neurons in the hidden layer and four sigmoid neurons in the output layer. Thus $2\times3+3\times4=18$ weights and $3+4=7$ biases must be determined.

Very small dimensions of the finally designed neural network with respect to similar networks is considerable: it has been reported in a similar application [5] 24 for the number of neurons in hidden layer, and in another paper [6] seven neurons. Compare these with final MLP neural network designed in this paper, which has only three neurons in hidden layer. Besides, it will be shown that this network has a good ability in learning, recalling and generalizing for new inputs.

Three training sets are learnt to the presented neural network: decision space samples with uniform distribution (Fig. 4 of Part 1), uniform samples which are mapped by complex conformal mapping (circular space in Fig. 9 of Part 1), and uniform equalized square space (Fig. 7 of Part 1). The last one is the data in Fig. 4 of Part 1, processed by the functions $\frac{0.25}{0.79}\times 4^{P}, \frac{0.25}{0.79}\times 4^{Q}$. The aim of the last mapping is equalizing the sizes of the classes and studying the effect of this mapping on learning ability of the neural network. From the theoretical view point, equalizing sizes of the classes may modify the learning of the network because it linearizes the classification space. In neural networks, which are based on a mathematical model, equalizing the sizes of the classes should have positive effect on learning of the network.

Experimental results show the opposite: the first series (non-equalized space) with 2520 iterations and the second series (equalized space by mapping) with 2750 iterations reach to the training error of $E_{Trn}=0.005$.

Despite of our conception, not only the performed mapping has not any positive effect on the learning of the network, but also it lengthens the learning procedure. This contradiction has been briefly discussed in Appendix 1.

Also the data having Gaussian distribution (Fig.s 5, 8, 10 of the Part 1) are tested on MLP network, but their learning was not as well as the data having uniform distribution. This shows that the MLP networks, opposite to the RBF networks, do not need to concentrate on class centers. It is better to use the data with uniform distribution, or preferably by a concentration on boundaries of the classes for MLPs. Briefly, *"MLP neural networks are bound oriented neural networks, and it must be focused on the bounds of the classes for training them"*.

About learning factor (or walking size) (α) and momentum (β) in error back propagation algorithm, it shall be noted that their selection and variations do not obey any known rule and it seems to have a fractal behavior. So, they have been varied adaptively in the neural network training computer program. α and β have had values between 0.9 to 1.1 and 0.7 to 0.95 respectively. If the momentum is selected much low, learning factor can not be increased very much, and neglecting this point may cause un-stability (divergence) problem in learning program. On the other hand, proper increase of the momentum can increase the inertia of movement toward global minimum. It rejects the effect of transient changes in movement trajectory.

Another important point is selection of error function norm [7], which has a significant role in network learning procedure. Note to this lemma that is proved mathematically in [8]:

Lemma 1: *Effect of the largest element of a vector on the higher order norms of the vector, is greater than its effect on the lower order norm. In other words, sensitivity of l-p norm with respect to the largest element of vector, increases by increasing the order of the norm i.e., p.*

This fact has been seen during the neural network learning: during the training procedure, several times has been seen that despite of decreasing error function norm, the maximum of error of samples was increasing. This problem generates bad estimations for samples related to the maximum error. In such a case, according to the above mentioned Lemma 1, the order of the error function must be increased. It concludes to decrease in both the error function and the samples maximum error. Certainly, increasing the order of error function norm can cause problems such as program divergence and floating point and overflow problems.

3 A Practical Case Study

Consider the three feeder distribution system of Fig. 2, having 16 switches and 13 line sections [9].

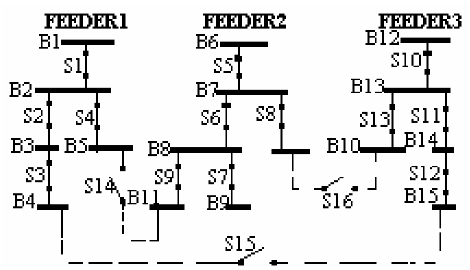

Fig. 2, Test distribution system

This system has been used for many distribution systems studied in literature, especially in IEEE Journals. Let us observe the performance of two intelligent optimizers, which have been designed in the previous sections. In this distribution system, each line section is known as a “zone” and a state is assigned for each zone as its load level. Thus the three feeder distribution system or as we call it “test distribution system” has a state vector containing 13 elements.

3.1 RBF Estimator Test

Gaussian (clustered our centered) distributed data is learnt better than uniformly distributed data. So, RBF neural network learnt by the Gaussian distributed data is tested as the final designed RBF neural estimator:

Consider two series of measurements for active and reactive powers (P and Q) in test distribution system (26 samples) that are shown in upper part of Fig. 3. None of these data have been learnt to the neural network. They are new inputs. The training set data are successfully taught to the neural network in learning mode. Thus, the previously learnt data are not tested in recalling mode, because they have been tested successfully in learning mode.

3.1.1 RBF Interpolation test:

The first series (the first 13 samples) belongs to the test distribution system operating in normal condition. As you see in Fig. 3, the RBF estimator in test for the first series has had three mistakes, which shows a 23% classification error. It is not a good estimation, so we can not be content with it.

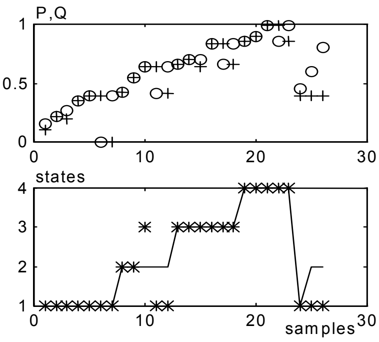

Fig. 3, RBF estimator test,
above: real power (+)and imaginary power (o),
below: desired states (solid line) and estimator output (*)

3.1.2 RBF Extrapolation test:

The second series belongs to operation of the test system in a faulted condition, such as: an unpredicted outage, increasing capacitance of the line and then abnormal change in the power factor of the line.

The data of this series are located out of the classes area. Thus, the estimator must estimate the most similar class (state) to the input data on the basis of “maximum likelihood” or “minimum distance”. This property is called as “Extrapolation”. The response of the neural RBF estimator is shown in lower graph of the Fig. 3. In this case, there has been two errors that both belong to contingency (faulted) condition, i.e. a 15% classification error. In latter section it will be shown that the MLP estimator has a more significant ability in this application.

3.2 MLP Estimator Test

Here, the MLP neural network, which is trained by the non-mapped uniformly distributed data, is tested as the best designed MLP neural network. For this purpose, two series of data that are measured in the test distribution system are used. Similar to the previous section, test data are thoroughly new and actual, and have not learned to the neural network in learning mode. So, the generalization ability of the network will be tested. The first series (the first 13 sample) belongs to normal operation of the distribution system, and the second series (the samples 14 to 26) belongs to a faulted condition. Inputs and outputs of the MLP estimator are shown in upper and lower section of the Fig.4, respectively.

3.2.1 MLP Interpolation test:

As you see in Fig. 4, the classification precision for the first series has been 100%, i.e. the classification error is 0%. It means that the MLP estimator has estimated the actual states (classes) for all of new inputs correctly.

3.2.2 MLP Extrapolation test:

About the second series: eight zones -out of 13 zones- have been affected by fault. Their power factors have exceeded from the allowed values. Thus, input values of the estimator are out of the classes and the estimator must estimate the nearest class to the inputs on the basis of maximum likelihood.

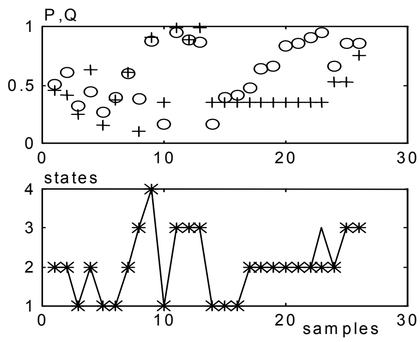

Fig. 4, MLP estimator test,
above: real power (+)and imaginary power (o),
below: desired states (solid line) and estimator output (*)

It is shown in Fig. 4, that all of estimations were correct, except one of them, i.e. a 93% classification precision or 07% classification error. The input, whose state has not been estimated correctly, is a critical case that is extra-ordinary, i.e. the case in which the power factor of the zone is changed 60%, abruptly. The distribution systems experts certify that the probability of such an abrupt change in power factor is very low.

The MLP neural network is an estimator which can perform maximum likelihood (or minimum distance) estimations for the data out of the classes (extrapolation) by a high performance. This property is resulted from the neural network generalization (intelligence) ability.

This important property of MLP estimator is very exciting and advantageous in special cases of the power systems such as contingency analysis, fault diagnosis and fault clearing. This ability has been reported for none of the similar systems in the literature [5,6]. Hence, you see that the MLP neural network, for the application of the decision support to utility management automation, acts very stronger than the RBF neural network from the both aspects of learning and generalization.

4 Conclusion

- A complete design procedure for two kinds of neural networks (MLP and RBF) with several training sets was presented and discussed on their various aspects.
- The finally designed MLP intelligent estimator does its job very well. It estimates the states of the test power system for two series of measured data. It is a good tool for utility management automation system as a decision support system (DSS). This subject has been shown in a practical example by comparing the estimated states and actual states and exposing their equality.
- The presented MLP neural network has the ability of maximum likelihood or minimum distance estimation for the data out of the classes (extrapolation). It is able to operate under the crises, fault, and contingency conditions. This property is the result of intelligence of artificial neural networks. It has not been reported in similar papers [5,6].
- The presented MLP neural network has much smaller dimensions than similar networks introduced in literature [5,6].
- Equalizing the classification space by nonlinear pre-processing may have not positive effect on MLP neural network training procedure. The reason may be disturbing the initial sensitivity factors, which may make training worse. This subject has been shown experimentally, and interpreted mathematically in the Appendix 1.
- The learning, recalling and generalizing (interpolation and extrapolation) ability of the MLP

networks is rather better than the RBF, for the classification application noted in this paper. In some other applications, it has been reported opposite of these results [3].

- Increasing the order of the network training error function (l-p norm) causes to have better control on the undesired increasing of maximum training error. It increases the sensitivity of the norm to the maximum training error [8]. On the other hand, it can make the training program unstable, or diverge. There is a trade off between these two factors: optimum control and stability of the program.
- Despite of circular shape of the Gaussian function cross section in RBF network neurons, transforming the rectangular classes to circular classes (by conformal mappings in complex plane), does not have any positive effects on the neural network training procedure. The reason of this may be disturbing the initial sensitivity factors and legalized form of original classification space.
- The positive effect of applying random noise to the training inputs of the RBF neural network by the "k-mean clustering technique" was experienced in two following cases:

 1. When the training and the error matrix calculation have had problems as divide by zero or ill conditioning.
 2. When the learning procedure has been very stupid or has been entrapped in local minima.

- The RBF neural network is a "Center Oriented" network from the view point of learning, recalling and generalizing.
- The MLP neural network is a "Bound Oriented" network from the view point of learning, recalling and generalizing.

References:

[1] Russel, S.J., Norvig, P., "*Artificial Intelligence: A Modern Approach*", ISBN: 964-377-008-7, 2003

[2] Chen, Z., "*Computational Intelligence for decision Support*" CRC Press LLC, 2000

[3] Hush, D.R., Horn, B.G., "Progress in Supervised Neural Networks", *IEEE Signal Processing Magazine,* Jan.1993

[4] Golden, R.M., "*Mathematical Methods for Neural Network Analysis and Design*", MIT Press, 1996

[5] Santoso, N.I. , Tan, O.T. , "Neural-Net Based Real-Time Control of Capacitors Installed on Distribution Systems" *IEEE Transactions on PWRD* , Vol.8 , No.4 , Oct. 1993

[6] Kim, H., Ko,H., Jung,K.H., "Artificial Neural-Network Based Feeder Reconfiguration for Loss Reduction in Distribution Systems" *IEEE Transactions on PWRD* , Vol.8 , No.3 July 1993

[7] Reddy, J.N.,"*Applied Functional Analysis and Variational Methods Engineering*" McGraw-Hill , 1986

[8] Fereidunian, A.R., Lesani, H., Lucas,C., "Distribution Systems Reconfiguration Using Pattern Recognizer Neural Networks" *International Journal of Engineering (IJE), Transactions B: Applications*, Vol. 15, No. 2 , pp. 135-144, July 2002

[9] Civanlar, S., Grainger, J.J., Yin, H., Lee, S.S.H., " Distribution Feeder Reconfiguration for Loss Reduction" *IEEE Transactions on PWRD* , Vol.3 , No.3 July 1988

Appendix 1. Discussion on Equalization Contradiction

Neural networks training, is performed by experiencing the training set, and consequently finding properties and boundaries of the classes by this kind of experience. So, processing the raw data, which are obtained from the basic fundaments that induce the classes, and is directly derived from the physical nature of the problem, may disturb the sensitivity factors. This probability will be higher, when the mapping function (processing) is non-linear.

Mathematically, for our special case, we have from the first order approximation of the MacLaurin's series:

$$\Delta f(p) = f(p+\Delta p) - f(p) \approx \frac{df(p)}{dp}.\Delta p \quad \text{(A.1)}$$

$$\Delta f(p) = (Ln4).(4^{p}).\Delta p \quad \text{(A.2)}$$

For example, for P1=0.2, P2 = 0.9 and ΔP = 0.01 :

$$\frac{\Delta f(P_1)}{\Delta f(P_2)} = \frac{0.483}{0.183} \cong 2.64 \quad \text{(A.3)}$$

You see, for an equal change of power in two various points, induced change in the mapped values is very different (by a 2.64 ratio). Meanwhile, if the mapping had been linear, this ratio should have been unity.

The reason of this is that the sensitivity factor is not constant. It is a linear function of p:

$$S_P^f = PLn4 = 1.3863P \quad \text{(A.4)}$$

So, sensitivity factor varies between 0 and 1.3863, when p varies between 0 and 1 per unit (normalized). This subject causes disturbance on the initial sensitivity factors of the classes with respect to state variables of the system. Consequently, the hidden logic, which underlies in the nature (physics) of classification problem, disturbs. For this reason, equalization of the classification space using a nonlinear mapping has made it harder. Further more, a [0 1] normalizing of mapped data increases the number of iterations to about 2800 for a training error of $E_{Trn} = 0.005$.