Input dimension reduction in neural network training-case study in transient stability assessment of large systems

S. Muknahallipatna

Badrul H. Chowdhury
Missouri University of Science and Technology, bchow@mst.edu

Follow this and additional works at: http://scholarsmine.mst.edu/faculty_work

Recommended Citation
http://scholarsmine.mst.edu/faculty_work/1196
INPUT DIMENSION REDUCTION IN NEURAL NETWORK TRAINING - CASE STUDY IN TRANSIENT STABILITY ASSESSMENT OF LARGE SYSTEMS

Suresh Muknahallipatna
Badrul H. Chowdhury

Abstract
The problem in modeling large systems by artificial neural networks (ANN) is that the size of the input vector can become excessively large. This condition can potentially increase the likelihood of convergence problems for the training algorithm adopted. Besides, the memory requirement and the processing time also increase. This paper addresses the issue of ANN input dimension reduction. Two different methods are compared for efficiency and accuracy when applied to transient stability assessment.

1. INTRODUCTION
On-line transient stability assessment (TSA) requires the identification of critical contingencies in a short enough time period so that the operator can be provided with the information as to whether the system will reach a stable state following a particular fault. The artificial neural network (ANN) implementation must be able to predict the trajectory of the system state following a disturbance, by using inputs obtained on-line at some particular instant of time.

A number of researchers have explored the possibility of using ANNs for the above tasks. Sharkawi, et al [1] used an ANN for transient stability assessment. A simple three generator power system was used for testing the network. A feedforward network with backpropagation training algorithm was used. Pao, et al [2] made an attempt to develop an ANN for predicting critical fault clearing times. A small power system consisting of four generators and seven lines was used. A total of thirty, twelve dimensional patterns were used to train the seven-neuron network. Sobajic and Pao [3] have used a combination of supervised and unsupervised learning for stability assessment. They have used a second order tensorial functional link model for unsupervised learning. This approach was demonstrated on a six-node, four machine power system. Fouad, et al [4] have applied a neural network technique to the concept of system vulnerability. They have tried to estimate the critical system parameter using a neural network.

Typically, when the size of the test system is small, the size of the neural network design is also small. Hence, the training required to determine synaptic weights of the network is fast, and convergence problems are less likely to occur given that the input features are selected with caution. However, with the increase in size of the power system, the number of neural network inputs also increases proportionately. This condition naturally increases the likelihood of training algorithm convergence problems. Besides, the memory requirement and the processing time have to be addressed as well. The purpose of this paper is therefore to address the issue of ANN input dimension reduction. Two different methods that exist in the literature are discussed.

The simple feedforward neural network with single hidden layer employing the backpropagation training algorithm (BPN)

Radial basis-function networks (RBFNs):

a) The Probabilistic Neural Network (PNN).

b) The General Regression Neural Network (GRNN).

A description of these networks and the rationale for choosing them are given in [5].

2. NETWORK ARCHITECTURE

The backpropagation, the probabilistic, and the general regression neural networks have been considered in this work for performing the task of stability classification. Specifically, these are:

1) Change in the rotor angle (from the pre-fault condition to the fault clearing time).  
2) Change in the angular velocity. 
3) Change in the terminal voltage. 
4) Change in generator real power. 
5) Change in generator reactive power.

Again, the suitability of these parameters for the task at hand have been presented in [5].

2.1 Feature Selection
The features selected to represent the inputs to the above ANNs were:

1) Change in the rotor angle (from the pre-fault condition to the fault clearing time).
2) Change in the angular velocity.
3) Change in the terminal voltage.
4) Change in generator real power.
5) Change in generator reactive power.

A description of the architectures used in the original ANN design is given first. Following that, the two methods of dimension reduction will be presented. Results of both the original and the modified networks will also be compared.

2.2 Training and Validation
The training and validation of the original neural network was done on the New England 39 bus test system and the IEEE 145 bus test system. The New England test system has ten generators and the IEEE test system has fifty generators. Only three-phase faults were considered. The faults were assumed to be cleared without any change in the network structure. The training data was generated using the digital simulation technique. Faults were created on the system, with the system prefault operating points at five different power levels for the New England system. For each fault, the system behavior at ten different fault clearing times considered around the critical clearing time for the fault was obtained. A total of 576 cases were simulated, with equal number of stable and unstable cases. Out of these, 73 cases were randomly selected for testing purposes. Each input pattern had a dimension of fifty since there are 10 generators in the system. On the other hand, a total of 898 cases were simulated for the IEEE test system. Out of these, 106 cases were randomly selected for testing. The training and the test patterns were simulated in the same way as was done for the New England test system. Each input pattern had a dimension of 250 since there are 50 generators in the system. The description of the neural networks used for the two test systems is presented in Table 1. The performance of the three types of networks studied is presented in Table 2. During the recall phase of the ANN, both the training and the testing sets were evaluated to test the network for generalization.

The RBPFNs have better performance compared to the BPN. This is expected, since they classify patterns by the nearest neighborhood criteria. The feedforward networks employing backpropagation training algorithm classify by finding decision surfaces. Since, the dimension of the input pattern is quite high, the
The process of finding the decision surfaces is complex. Therefore, those networks employing the backpropagation training algorithm exhibit poorer performance when compared to the RBFNs.

### Table 1. Structure of the original neural networks.

<table>
<thead>
<tr>
<th>NETWORK</th>
<th>NEW ENGLAND TEST SYSTEM</th>
<th>IEEE TEST SYSTEM</th>
</tr>
</thead>
<tbody>
<tr>
<td># of neurons in Input layer</td>
<td>50</td>
<td>250</td>
</tr>
<tr>
<td># of neurons in Hidden layer</td>
<td>25</td>
<td>154</td>
</tr>
<tr>
<td># of neurons in Output layer</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### Table 2. Performance of the original neural networks.

<table>
<thead>
<tr>
<th>NETWORK</th>
<th>NEW ENGLAND TEST SYSTEM</th>
<th>IEEE TEST SYSTEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stable cases classified as Unstable</td>
<td>10</td>
<td>27</td>
</tr>
<tr>
<td>Unstable cases classified as Stable</td>
<td>8</td>
<td>23</td>
</tr>
<tr>
<td>True Classifications</td>
<td>558</td>
<td>848</td>
</tr>
</tbody>
</table>

### 3. DIMENSION REDUCTION

The dimension of the input data for the neural network model developed in Section 2 is fixed by the number of generators in a power system. This dimension, for the New England test system, was 50, and in case of the IEEE test system, it was 250. The present day computers with their memory and CPU speed capabilities can handle the dimension of a system similar to the New England test system with considerable ease. However, in case of the IEEE test system or larger systems, the processing time and the memory requirements increase significantly. In addition to the memory requirement due to the large dimension of the input data, the memory requirements of the RBFNs are increased due to storage of the input patterns. One can reduce the memory requirement and processing time, by reducing, (i) the dimension of the input data, or (ii) the number of input patterns (training patterns). In this section, the problem of reducing the dimension of the input data will be addressed.

The aim of dimension reduction is to describe the input patterns by means of a minimum number of features which are effective in discriminating between different classes. Most of the dimension reduction (also called feature reduction) methods are classified into two groups [6]:

- **Subsetting methods.**
- **Feature space transformation methods.**

#### Subsetting methods

These methods are also known as filtering methods. In these methods, the dimension is reduced by selecting a few of the original features and ignoring the others. The selection process is usually done by considering the following principles:

- Only input features having an effect on the output are selected.
- Input features having the same information are represented by a single input feature.

A statistical measure such as the linear correlation method, can be used to implement the above principles.

#### Feature space transformation methods

These methods are also known as aggregation methods. In these methods, the dimension of the sample input space is reduced by constructing a new set of features in a lower dimensional space. The new set of features can be a linear or a non-linear combination of the original features. A number of methods like Karhunen-Loeve [7] transformation, divergence method [8], non-parametric discriminant analysis [9], discriminant analysis [9] etc., are available to perform the transformation. In this work, the discriminant analysis method has been used to reduce the feature space dimension.

#### 3.1 Statistical Correlation Technique

As mentioned earlier, the linear correlation between the input variables is computed. If the correlation between the ith and the jth variable of a machine is greater than or equal to 0.9, then one of the variables is ignored or discarded in representing the input data. This method was first applied to the input data of the New England Test System.

#### 3.1.1 Results for the New England Test System

The correlation technique was applied to subsets of the input variables. The input variables were grouped into five subsets. Each subset contained the same type of variables, i.e., all rotor angles of the generators, and so on. Using the computer package Matlab™, the linear correlation coefficients between the variables in each subset were computed. The correlation coefficients for the rotor angle variable is shown in Table 3.

### Table 3. Linear correlation coefficient between the rotor angles.

<table>
<thead>
<tr>
<th>Generators</th>
<th>30</th>
<th>31</th>
<th>32</th>
<th>33</th>
<th>34</th>
<th>35</th>
<th>36</th>
<th>37</th>
<th>38</th>
<th>39</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>1</td>
<td>0.80</td>
<td>0.80</td>
<td>0.90</td>
<td>0.91</td>
<td>0.90</td>
<td>0.89</td>
<td>0.92</td>
<td>0.89</td>
<td>0.78</td>
</tr>
<tr>
<td>31</td>
<td>0.80</td>
<td>1</td>
<td>0.31</td>
<td>0.04</td>
<td>0.03</td>
<td>0.01</td>
<td>0.04</td>
<td>0.33</td>
<td>0.70</td>
<td>0.09</td>
</tr>
<tr>
<td>32</td>
<td>0.80</td>
<td>0.31</td>
<td>1</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
<td>0.84</td>
<td>0.79</td>
<td>0.72</td>
<td>0.69</td>
</tr>
<tr>
<td>33</td>
<td>0.90</td>
<td>0.04</td>
<td>0.85</td>
<td>1</td>
<td>0.97</td>
<td>0.99</td>
<td>0.99</td>
<td>0.90</td>
<td>0.85</td>
<td>0.76</td>
</tr>
<tr>
<td>34</td>
<td>0.91</td>
<td>0.03</td>
<td>0.85</td>
<td>0.97</td>
<td>1</td>
<td>0.98</td>
<td>0.97</td>
<td>0.90</td>
<td>0.85</td>
<td>0.79</td>
</tr>
<tr>
<td>35</td>
<td>0.90</td>
<td>0.01</td>
<td>0.85</td>
<td>0.99</td>
<td>0.98</td>
<td>1</td>
<td>0.99</td>
<td>0.90</td>
<td>0.85</td>
<td>0.76</td>
</tr>
<tr>
<td>36</td>
<td>0.89</td>
<td>0.04</td>
<td>0.84</td>
<td>0.99</td>
<td>0.97</td>
<td>0.99</td>
<td>1</td>
<td>0.88</td>
<td>0.84</td>
<td>0.75</td>
</tr>
<tr>
<td>37</td>
<td>0.92</td>
<td>0.33</td>
<td>0.79</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
<td>0.88</td>
<td>1</td>
<td>0.91</td>
<td>0.69</td>
</tr>
<tr>
<td>38</td>
<td>0.89</td>
<td>0.30</td>
<td>0.72</td>
<td>0.85</td>
<td>0.85</td>
<td>0.83</td>
<td>0.84</td>
<td>0.91</td>
<td>1</td>
<td>0.74</td>
</tr>
<tr>
<td>39</td>
<td>0.78</td>
<td>0.09</td>
<td>0.69</td>
<td>0.76</td>
<td>0.79</td>
<td>0.76</td>
<td>0.75</td>
<td>0.69</td>
<td>0.74</td>
<td>1</td>
</tr>
</tbody>
</table>

As mentioned before, the criteria to discard an input variable is that, if two variables are correlated (≥ 0.9), then discard one of the variables. The criteria fails in indicating which variable out of the two should be discarded. This process of discarding becomes more complicated when a number of variables are correlated to each other. Table 3 indicates that the rotor angles of generators
30, 33, 34, 35, and 37 are highly correlated. It is not possible to decide which of the above rotor angles should be discarded using the correlation criteria, since it does not indicate which inputs have better information about class separability. In order to overcome this problem, it is necessary to consider whether a variable selected as a feature will provide more information for classification than those not selected. This information is usually obtained by considering the heuristic notion of interclass distance.

**Interclass Distance** [10]

Given a set of patterns with dimension n, it is reasonable to assume that the pattern vectors for each of the two classes occupy a distinct region in the observation space [11]. The average pairwise distance between the patterns is a measure of class separability in the region with respect to a particular variable. This measure of class separability for an ith variable is given by Egn. (1), as follows:

\[ F_i = \frac{m_i^S - m_i^U}{\sigma_i^S + \sigma_i^U} \]

for \( i = 1 \) to \( n \). \hspace{1cm} (1)

where:

- \( m_i^S \) and \( \sigma_i^S \) are the mean and variance respectively of the ith variable corresponding to the stable class; and
- \( m_i^U \) and \( \sigma_i^U \) are the mean and variance respectively of the ith variable corresponding to the unstable class.

The variables having the higher value of index \( F \) carry more information about class separability. The index \( F \) for each variable is shown in Table 4. Using this interclass distance measure with the linear correlation coefficients for the variables, the input variables to be discarded were selected. For example, in Table 3, the correlation coefficients between the rotor angles of generators 30, with those of 33, 34, 35, and 37 are high. Comparing, the corresponding \( F_i \) values of the rotor angles for these generators given in Table 4, it can be seen that the rotor angle parameter of the generator 35, has the highest value of \( F \). This indicates that only the rotor angle variable corresponding to generator 35 should be retained and the rest discarded. Continuing on in this way, the next row of Table 3 shows no correlation of generator 31 (which was not discarded in the previous step) with any of the other generators. Therefore generator 31 cannot be discarded. The technique of discarding proceeds in this way. Table 5 shows the input variables discarded for each of the generators.

**Table 4. Interclass distance of the input variables for the New England test system.**

<table>
<thead>
<tr>
<th>Gen</th>
<th>Rotor angle</th>
<th>Angular velocity</th>
<th>Terminal voltage</th>
<th>Real power</th>
<th>Reactive power</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>0.1026</td>
<td>0.2994</td>
<td>0.4247</td>
<td>0.1776</td>
<td>0.4277</td>
</tr>
<tr>
<td>31</td>
<td>0.2061</td>
<td>0.2693</td>
<td>0.3370</td>
<td>0.1395</td>
<td>0.3244</td>
</tr>
<tr>
<td>32</td>
<td>0.2303</td>
<td>0.2743</td>
<td>0.3522</td>
<td>0.0853</td>
<td>0.3163</td>
</tr>
<tr>
<td>33</td>
<td>0.2095</td>
<td>0.3079</td>
<td>0.4824</td>
<td>0.0206</td>
<td>0.4150</td>
</tr>
<tr>
<td>34</td>
<td>0.2113</td>
<td>0.3652</td>
<td>0.4355</td>
<td>0.3345</td>
<td>0.6228</td>
</tr>
<tr>
<td>35</td>
<td>0.2295</td>
<td>0.3392</td>
<td>0.4728</td>
<td>0.1158</td>
<td>0.4708</td>
</tr>
<tr>
<td>36</td>
<td>0.2397</td>
<td>0.3182</td>
<td>0.4518</td>
<td>0.0279</td>
<td>0.4139</td>
</tr>
<tr>
<td>37</td>
<td>0.2049</td>
<td>0.2482</td>
<td>0.3914</td>
<td>0.0506</td>
<td>0.2132</td>
</tr>
<tr>
<td>38</td>
<td>0.2907</td>
<td>0.5356</td>
<td>0.4975</td>
<td>0.0698</td>
<td>0.3714</td>
</tr>
<tr>
<td>39</td>
<td>0.0214</td>
<td>0.1088</td>
<td>0.3553</td>
<td>0.2941</td>
<td>0.2835</td>
</tr>
</tbody>
</table>

In Table 5, the 'X' mark indicates a variable that is discarded. After these input variables are ignored, the new dimension of the input patterns reduces from 50 to 31. When singular value decomposition is performed on the weight matrix, small singular values indicate that their corresponding inputs have the least effect on the performance of the network.

**Training and Testing of the Modified Neural Network**

Using the reduced input vectors, the same three networks, with modification in their structure to suit the new input dimension, were trained and tested. The description of the modified neural networks is presented in Table 6 and the performance of these networks is presented in Table 7. Comparing Table 6 with Table 1, one can notice that the number of hidden layer neurons have increased for the modified case. Generally, a network having a higher number of weights, as a result of higher number of neurons in the hidden layer, has more degrees of freedom leading to an unconstrained network. The generalization error of an unconstrained network is high. On the other hand, a smaller network (highly constrained) will be sensitive to initial conditions and learning parameters. It may get stuck at a local minima due to an unfavorable set of initial conditions. In order to avoid these problems, an optimal number of hidden layer neurons was determined for the modified network, which happens to higher than that for the original ANN.

**Table 5. Discarded input variables.**

<table>
<thead>
<tr>
<th>Gen</th>
<th>Rotor angle</th>
<th>Angular velocity</th>
<th>Terminal voltage</th>
<th>Real power</th>
<th>Reactive power</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>36</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>37</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>38</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>39</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 6. Structure of the Modified ANNs**

<table>
<thead>
<tr>
<th>New England Test System</th>
<th>Network</th>
<th>BPN</th>
<th>PNN</th>
<th>GRNN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># of neurons in Input layer.</td>
<td>31</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td># of neurons in Hidden layer</td>
<td>37</td>
<td>503</td>
<td>503</td>
</tr>
<tr>
<td></td>
<td># of neurons in Output layer</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

**Table 7. Comparative performance results for the Modified ANNs**

<table>
<thead>
<tr>
<th>New England Test System</th>
<th>NETWORK</th>
<th>BPN</th>
<th>PNN</th>
<th>GRNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stable cases classified as Unstable</td>
<td>7 (10)</td>
<td>13 (2)</td>
<td>2 (2)</td>
<td></td>
</tr>
<tr>
<td>Unstable cases classified as Stable</td>
<td>8 (8)</td>
<td>26 (14)</td>
<td>1 (1)</td>
<td></td>
</tr>
<tr>
<td>True</td>
<td>561</td>
<td>537</td>
<td>573</td>
<td></td>
</tr>
<tr>
<td>Classifications</td>
<td>97.39%</td>
<td>93.22%</td>
<td>99.47%</td>
<td></td>
</tr>
</tbody>
</table>

In Table 7, the numbers in parentheses represent the results obtained using the original input data set. It can be seen that the same level of performance has been maintained by the backpropagation NN and the GRNN, whereas the performance level of the PNN has deteriorated. This behavior is due to the following reasons:

- The PNN uses the Parzen probability density function estimator employing a Gaussian kernel. This type of PDF requires the use of Patrick-Fisher separability measures [10] for proper classification.
- The interclass distance measure given by Egn. (1), used to select the features does not estimate the probability density functions.
3.1.2 Results for the IEEE test system

As a first step, the linear correlation coefficients of the input variables were computed. But, it was observed that only a few input variables (compared to the original input dimension) had significant correlation. The reason behind this lies in the fact that this power system exhibits inter-area mode instability in addition to the regular local instability for different faults. The inter-area mode instability is characterized by a group of generators swinging against another group of generators following a disturbance. In case of the local instability, a set of generators will swing against another set of generators and both sets could belong to a single area. Since, both these modes of instability are present in this system, the parameters that are chosen as input variables to the neural network will not have significant correlation.

3.2 Discriminant Analysis

Discriminant analysis is one of the well known linear feature extraction techniques. In this method, the input patterns in the original pattern space are projected into a new subspace having fewer dimensions than the original pattern space. Mathematically this can be written as:

$$Y_j = T_{0j} X_j \quad \forall j = 1, \ldots, n \text{ and } i = 1, \ldots, K$$  \hspace{1cm} (2)

where

- $Y$ denotes the patterns in the reduced pattern space, and
- $X$ denotes the patterns in the original pattern space, and
- $K$ denotes the number of classes, and
- $T_0$ denotes the transformation matrix.

The process of projection into the subspace or constructing the transformation matrix $T_0$ has to satisfy the following constraints:

- the ratio of the between-class scatter to the within-class scatter should be maximum.
- the number of eigenvectors chosen will decide the dimension of the new pattern space. In the technique discussed in reference [9] this number is decided by satisfying the constraint of the number of pattern classes.

3.2.1 Computation of the Projection Matrix $T_0$

Let the patterns in the original space be described by $d$-dimensional patterns and be separated into $K$ classes. Let the unnormalized patterns in the $k$th class be represented by the column vectors given below:

$$[x_{1k}^*, x_{2k}^*, \ldots, x_{nk}^*)^T]$$

where

$$x_{ij}^k = \left[ x_{ij1}^k, x_{ij2}^k, \ldots, x_{ijd}^k \right]^T$$

$n_k$ = number of patterns in the $k$th class.

superscript * denotes that the patterns are unnormalized.

d = dimension of the patterns.

**STEP1:** Compute the mean of the $i$th feature for the $k$th class.

$$m_k = \frac{1}{n_k} \sum_{j=1}^{n_k} x_{jik}^k$$  \hspace{1cm} (6)

**STEP2:** Compute the vector of feature means for the $k$th class.

$$m_k = [m_{k1}, m_{k2}, \ldots, m_{kd}]^T$$  \hspace{1cm} (7)

**STEP3:** Compute the pooled mean or the grand mean vector for all the patterns using:

$$m = \frac{1}{n} \sum_{k=1}^{K} n_k m_k$$  \hspace{1cm} (8)

where

$$n = \sum_{k=1}^{K} n_k$$

**STEP4:** The scatter matrix $S$ for the $k$th class is defined by:

$$S_k^w = \sum_{j=1}^{n_k} (x_{jik}^k - m_k)(x_{jik}^k - m_k)^T$$  \hspace{1cm} (9)

**STEP5:** Compute the within-class scatter matrix, $S^w$, as the sum of the class scatter matrices:

$$S^w = \sum_{k=1}^{K} S_k^w$$  \hspace{1cm} (10)

**STEP6:** Compute the between-class scatter matrix, $S^b$, as the scatter matrix for the class means using:

$$S^b = \sum_{k=1}^{K} n_k (m_k - m)(m_k - m)^T$$  \hspace{1cm} (11)

**STEP7:** Form the matrix $P$, using Eqn. (5).

**STEP8:** Compute the eigenvalues and the eigenvectors of the matrix $P$. The same level of information present in the input patterns in the original pattern space has to be maintained in the new pattern space also. Therefore, an optimum number of eigenvectors corresponding to non-zero eigenvalues has to be chosen to construct the projection matrix $T_0$. The number of eigenvectors chosen will decide the dimension of the new pattern space. In the technique discussed in reference [9] this number is decided by satisfying the constraint of

$$\lambda_j \geq \lambda_{j+1} \quad j = 1, \ldots, m-1$$

where

$$\lambda_j = \text{eigenvalues}$$

The eigenvectors corresponding to the $m-1$ non-zero eigenvalues are considered in constructing the projection matrix, where $m$ is the number of pattern classes. Foley, et al. [12] have developed an optimal set of discriminant vectors. In their work, the first feature is the Fisher discriminant vector. The second feature is found by maximizing the Fisher criterion subject to the constraint that the second feature be orthogonal to the Fisher discriminant vector. Okada, et al. [13] have proposed the orthonormal discriminant vector method. In this method, a maximum of $n-1$ features can be extracted, where $n$ is the dimension of the original pattern space. In this paper, the version presented in reference [8] has been used.

**STEP9:** Form the matrix $Q$, using Eqn. (5).

**STEP10:** Compute the eigenvalues and the eigenvectors of the matrix $Q$. The same level of information present in the input patterns in the original pattern space has to be maintained in the new pattern space also. Therefore, an optimum number of eigenvectors corresponding to non-zero eigenvalues has to be chosen to construct the projection matrix $T_0$. The number of eigenvectors chosen will decide the dimension of the new pattern space. In the technique discussed in reference [9] this number is decided by satisfying the constraint of

$$\lambda_j \geq \lambda_{j+1} \quad j = 1, \ldots, m-1$$

where

$$\lambda_j = \text{eigenvalues}$$

The eigenvectors corresponding to the $m-1$ non-zero eigenvalues are considered in constructing the projection matrix, where $m$ is the number of pattern classes. Foley, et al. [12] have developed an optimal set of discriminant vectors. In their work, the first feature is the Fisher discriminant vector. The second feature is found by maximizing the Fisher criterion subject to the constraint that the second feature be orthogonal to the Fisher discriminant vector. Okada, et al. [13] have proposed the orthonormal discriminant vector method. In this method, a maximum of $n-1$ features can be extracted, where $n$ is the dimension of the original pattern space. In this paper, the version presented in reference [8] has been used.

**STEP11:** Form the matrix $D$, using Eqn. (5).

**STEP12:** Compute the eigenvalues and the eigenvectors of the matrix $D$. The same level of information present in the input patterns in the original pattern space has to be maintained in the new pattern space also. Therefore, an optimum number of eigenvectors corresponding to non-zero eigenvalues has to be chosen to construct the projection matrix $T_0$. The number of eigenvectors chosen will decide the dimension of the new pattern space. In the technique discussed in reference [9] this number is decided by satisfying the constraint of

$$\lambda_j \geq \lambda_{j+1} \quad j = 1, \ldots, m-1$$

where

$$\lambda_j = \text{eigenvalues}$$

The eigenvectors corresponding to the $m-1$ non-zero eigenvalues are considered in constructing the projection matrix, where $m$ is the number of pattern classes. Foley, et al. [12] have developed an optimal set of discriminant vectors. In their work, the first feature is the Fisher discriminant vector. The second feature is found by maximizing the Fisher criterion subject to the constraint that the second feature be orthogonal to the Fisher discriminant vector. Okada, et al. [13] have proposed the orthonormal discriminant vector method. In this method, a maximum of $n-1$ features can be extracted, where $n$ is the dimension of the original pattern space. In this paper, the version presented in reference [8] has been used.
maintaining 95% of the variance present in the original pattern space. Mathematically, it can be represented by:

\[ r_m = \frac{m}{d} \sum_{i=1}^{d} \lambda_i \geq 0.95 \]  

where:
- \( m \) represents the dimension in the new pattern space.
- \( \lambda_i \) are the eigenvalues of the matrix \( P \).

**STEP9:** Construct the projection matrix \( T_o \) using the selected eigenvectors of matrix \( P \) as the columns.

**STEP10:** Finally, compute the input patterns in the new pattern space using Eqn. (2).

### 3.2.2 Training and Validation

The discriminant analysis method was implemented using the computer package Matlab \( \text{TM} \). Using this program, the input patterns in the original pattern space of dimension 50 were transformed to a new pattern space of dimension 15 in case of the New England test system, and from a dimension of 250 to a new dimension of 80 for the IEEE test system. Since the input variables have been transformed into a new subspace, the new variables have no one-to-one correspondence to physical parameters as observed in the original network.

Using these new patterns, the three neural networks considered in the previous section were constructed for each of the above test systems. The networks were modified to suit the new input pattern dimension. The description of the neural networks developed is presented in Table 8. The performance of these three types of networks studied is presented in Table 9.

### 4. CONCLUSIONS

The discriminant analysis method has shown that even with reduced dimension of the input pattern space, the same or better level of classification can be maintained with suitable neural networks. As demonstrated, even though the subsetting method is simple to implement, it is dependent on the behavior of the power system to disturbances. Thus, the discriminant analysis method appears to be superior for input dimension reduction in modeling large systems by neural networks.

### 5. REFERENCES