6.1 INTRODUCTION

In the previous chapters, the network was trained with the 17-dimensional input vector. The input layer of the network was presented with 17 features. This chapter presents conversion of the 17 dimensions into 2 dimensions. The 2-dimensional vector is presented in the input layer of the network. Because of this, the size of the network is reduced from 17 nodes to 2 nodes in the input layer. In order to convert 17 inputs into 2 inputs, the 17 dimensional vectors are mapped onto a 2-dimensional space, by using a transformation. The transformed 2-dimensional vector does not represent any individual feature; instead, it is a combination of 17 features with no dimensional quantity.

6.2 PRINCIPLE OF TRANSFORMATION

The transformation of a set on n-dimensional real vectors onto a plane is called a mapping operation. The result of this operation is a planar display. The mapping operation can be linear or non-linear. Fisher (1936)
has developed a linear classification algorithm. Hong and Yang (1991) have used a method for constructing a classifier on the optimal discriminant plane, by using minimum distance criterion for multiclass classification for less number of patterns. Foley (1972) had discussed the method of considering number of patterns and feature size. Gallinari, Thiria, Bardan and Soulie (1991) have discussed the relations between discriminant analysis and multiplayer perceptrons. A linear mapping is used to map a n-dimensional vector space, \( \mathbb{R}^n \), on to a two-dimensional space. Many methods are available under both linear and non-linear mappings. Siedlecki, Siedlecka and Skalansky (1988) have given an overview of mapping techniques. The mapping of the original vector ‘\( X \)’ onto a new vector ‘\( Y \)’ on a plane is done by a matrix transformation, which is given by;

\[
Y = AX + b
\]

\[\text{...(6.2.1)}\]

where

\[
A = \begin{bmatrix}
\phi_1^T \\
\ldots \\
\phi_n^T
\end{bmatrix}
\]

\[\text{...(6.2.2)}\]

\( b \) is a 2-dimensional vector, 
\( \phi_1 \) is a projection vector (also called a discriminant vector), and 
\( \phi_2 \) is another projection vector (also called a discriminant vector).
The 2-dimensional vector ‘b’ does not introduce any relevant information, but it is given for the generality of the equation 6.2.1. The steps involved in the linear mappings are:

**Step 1:** Computation of the discriminant vectors $\varphi_1$ and $\varphi_2$: this is specific for a particular linear mapping algorithm.

**Step 2:** Computation of the planar images of the original data points: this is common for all linear mapping algorithms.

Some of the linear mapping algorithms are:

i) Principal component mapping, Kittler and Young (1973)


iii) Least squared error mapping, Mix and Jones (1982) and

iv) Projection pursuit mapping, Friedman and Turkey (1974)

Out of the above four methods, generalised declustering optimal discriminant plane, which is based on a mapping technique of Fisher, is chosen. The vectors $\varphi_1$ and $\varphi_2$ are discriminant vectors, and the plane formed by them is the discriminant plane, which is optimal.
6.3 COMPUTATION OF DISCRIMINANT VECTORS $\varphi_1$ AND $\varphi_2$

This computation of discriminant vectors $\varphi_1$ and $\varphi_2$ has been done by Hussan J. H. and Purushothaman S. (2006) [55] which is presented here for ready reference.

The Fisher's criterion is given by:

$$J(\varphi) = \frac{\varphi^T S_b \varphi}{\varphi^T S_w \varphi} \quad \ldots(6.3.1)$$

where

$S_b$ is the between class matrix, and
$S_w$ is the within class matrix, which is non-singular.

$$S_b = \sum_{i=1}^{m} P(\omega_i)(m_i - m_0)(m_i - m_0)^T \quad \ldots(6.3.2)$$

$$S_w = \sum_{i=1}^{m} P(\omega_i)E[(x_i - m_i)(x_i - m_i)^T / \omega_i] \quad \ldots(6.3.3)$$

where

$P(\omega_i)$ is a priori the probability of the $i^{th}$ pattern, generally, $P(\omega_i) = 1/m$,
$m_i$ is the mean vector of the $i^{th}$ class patterns, $i=1,2,\ldots,m$;
$m_o$ is the global mean vector of all the patterns in all the classes.

$X = \{x_i, i=1,2,\ldots,L\}$ is the $n$-dimensional pattern of each class.

The discriminant vector, that maximizes $J$ in equation 6.3.1, is denoted by $\varphi_1$.

The vector $\varphi_1$ is found as a solution of the eigenvalue problem given by:

$$S_b \varphi_1 = \lambda m_1 S_w \varphi_1 \quad \ldots(6.3.4)$$
where $\lambda_{m1}$ is the greatest non-zero eigenvalue of $S_b S_w^{-1}$. The eigenvector corresponding to $\lambda_{m1}$ is $\varphi_1$. Another discriminant vector $\varphi_2$ is obtained by using same criterion of equation 6.3.1. The vector $\varphi_2$ should also satisfy the equation given by:

$$\varphi_2^T \varphi_1 = 0.0 \quad ...(6.3.7)$$

The equation 6.3.4 indicates, that the solution obtained is geometrically independent. The discriminant vector $\varphi_2$ is found as a solution of the eigenvalue problem, which is given by:

$$Q_p S_b \varphi_2 = \lambda_{m2} S_w \varphi_2 \quad ...(6.3.8)$$

where $\lambda_{m2}$ is the greatest non-zero eigenvalue of $Q_p S_b S_w^{-1}$, and $Q_p$ is the projection matrix given by:

$$Q = I - \frac{\varphi_1 \varphi_1^T S_w^{-1}}{\varphi_1^T S_w^{-1} \varphi_1} \quad ...(6.3.9)$$

where $I$ is an identity matrix.

In equation (6.3.4 and 6.3.8), $S_w$ should be non-singular. It is absolutely necessary, that $S_w$ should be non-singular, even for a more general discriminating analysis and generating multi orthonormal vectors, according to [Foley and Sammon (1975), Liu, Cheng and Yang (1992), and Cheng, Zhuang and Yang (1992)]. If $S_w$ is singular, $S_w$ should be made non-singular, by using singular value decomposition (SVD) method and
perturbing the matrix. Klema and Laub (1980) have discussed the SVD and the method of its computation with some applications. Sullivan and Liu (1984) have explained the SVD with application to signal processing. By using equations (6.3.4 and 6.3.8), the values of $\varphi_1$ and $\varphi_2$ discriminant vectors are obtained.

6.4 COMPUTATION OF 2-DIMENSIONAL VECTOR FROM THE ORIGINAL N-DIMENSIONAL VECTOR

The 2-dimensiononal vector set are denoted by $y_i$. The vector $y_i$ is given by:

$$y_i = (u_i, v_i) = \begin{bmatrix} X_i^T \varphi_1 \quad X_i^T \varphi_2 \end{bmatrix} \quad \ldots(6.4.1)$$

The vector set $y_i$, is obtained by projecting the original vector 'X' of the patterns onto the space spanned by $\varphi_1$ and $\varphi_2$ by using equation 6.4.1.

Let $X^{(k)} \subset X$ belong to the $K^{th}$ class. The corresponding 2-dimensional set of vectors are:

$$y^{(k)} = (y_1^{(k)}, y_2^{(k)}, \ldots, y_{L_k}^{(k)}) \quad \ldots(6.4.2)$$

$$\sum_{i=1}^{m} L_k - L$$

where

$$y_i^{(k)} = (u_i^{(k)}, v_i^{(k)}), i = 1, 2, 3, \ldots, L_k$$

The 2-dimensional vectors of the training patterns are given in Table 6.1

82
6.5 PROCEDURE FOR IMPLEMENTING OPTIMAL DISCRIMINANT PLANE METHOD IN NEURAL NETWORK

The flow-chart for implementing the optimal discriminant plane method in neural network is given in the Figure 6.1. The patterns given in Table 2.1 are normalized and selected for training the network. The within class matrix 'Sw' and between class matrix 'Sb' are calculated. The 'Sw' matrix is checked for non-singularity. If 'Sw' matrix is singular, singular value decomposition is applied to 'Sw' and small perturbation is done. After perturbation, 'Sw' matrix is recomputed. By using equations (6.3.4 and 6.3.8), the φ₁ and φ₂ discriminant vectors are obtained. By using φ₁ and φ₂ vectors, the normalized input patterns are transformed to a 2-dimensional vector. The 2-dimensional vectors are given in Table 6.1. The 2-dimensional vector is used as the input to the network and the outputs of the nodes in the output layer are calculated. Based upon the type of weight updating algorithm, the network is trained to learn all the 2-dimensional vectors, and the training of the Network is stopped when a performance index of the network is reached. For testing the network’s classification performance, the test patterns are transformed to 2-dimensional vectors and used as inputs to the network.
Fig. 6.1 Flow-chart for implementing optimal discriminant plane method in Neural network

Start

Read patterns

Normalize the patterns

Select the patterns

Calculate $S_b$ and $S_w$ matrices

Is $S_w$ singular?

Yes

Convert $S_w$ into non-singular using Singular Value Decomposition (SVD)

No

Calculate $\phi_1$ and $\phi_2$ vectors

Input transformed vectors into the network

Train the network with BPA / EKF algorithms

Test the network with BPA / EKF algorithms

Stop

Fig. 6.1 Flow-chart for implementing optimal discriminant plane method in Neural network
Table 6.1 The 2-dimensional vectors of the original 17-dimensional input patterns.

<table>
<thead>
<tr>
<th>Sample Pattern No.</th>
<th>U</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.248405</td>
<td>0.113239</td>
</tr>
<tr>
<td>2</td>
<td>0.284535</td>
<td>0.146954</td>
</tr>
<tr>
<td>3</td>
<td>0.286177</td>
<td>0.157134</td>
</tr>
<tr>
<td>4</td>
<td>0.075246</td>
<td>-0.138888</td>
</tr>
<tr>
<td>5</td>
<td>0.126176</td>
<td>-0.220629</td>
</tr>
<tr>
<td>6</td>
<td>0.309048</td>
<td>0.021163</td>
</tr>
<tr>
<td>7</td>
<td>0.208151</td>
<td>-0.106131</td>
</tr>
<tr>
<td>8</td>
<td>0.377304</td>
<td>0.118302</td>
</tr>
<tr>
<td>9</td>
<td>0.033771</td>
<td>-0.257652</td>
</tr>
<tr>
<td>10</td>
<td>0.064219</td>
<td>-0.263593</td>
</tr>
<tr>
<td>11</td>
<td>0.427905</td>
<td>0.302125</td>
</tr>
<tr>
<td>12</td>
<td>0.296603</td>
<td>-0.172194</td>
</tr>
<tr>
<td>13</td>
<td>0.429281</td>
<td>0.293695</td>
</tr>
<tr>
<td>14</td>
<td>0.311602</td>
<td>-0.200051</td>
</tr>
<tr>
<td>15</td>
<td>0.344748</td>
<td>-0.018211</td>
</tr>
<tr>
<td>16</td>
<td>0.283635</td>
<td>-0.140848</td>
</tr>
<tr>
<td>17</td>
<td>0.296513</td>
<td>-0.163203</td>
</tr>
<tr>
<td>18</td>
<td>0.123165</td>
<td>-0.151501</td>
</tr>
<tr>
<td>19</td>
<td>0.136756</td>
<td>-0.104236</td>
</tr>
<tr>
<td>20</td>
<td>0.010996</td>
<td>-0.170528</td>
</tr>
<tr>
<td>21</td>
<td>-0.036130</td>
<td>-0.128443</td>
</tr>
<tr>
<td>22</td>
<td>-0.205433</td>
<td>-0.199332</td>
</tr>
<tr>
<td>23</td>
<td>-0.375849</td>
<td>-0.230142</td>
</tr>
<tr>
<td>24</td>
<td>-0.389305</td>
<td>-0.451422</td>
</tr>
</tbody>
</table>
6.5.1 RESULTS OF ANN TRAINED BY USING BPA WITH TRANSFORMED INPUT VECTORS

The network is trained by using BPA. The original input patterns are transformed using $\varphi_1$ and $\varphi_2$ vectors. The value of $\eta$ used is 1 and $\alpha$ used is 0.8. The configurations of the network are 2-6-1. Test patterns are transformed into 2-dimensional vectors and presented to the network at the end of every iteration to evaluate the classification performance of the network. A maximum classification performance of 96.38% is obtained in 9 iterations at MSE of 0.2732, whereas the network trained with transformation of input vectors requires 19 iterations (Figure 2.8) to reach a classification performance of 96.38%. In addition to the reduction in the number of iterations to reach the maximum classification performance, the size of the network is reduced and hence, the number of arithmetic operations is reduced. Training the network beyond 9th iteration results only
in slight improvement in the classification performance. The classification performance and the MSE curves for the network are shown in Figure 6.2.

Fig.6.2 MSE and Classification Performance of the network trained by using BPA with transformed input vectors
6.5.2 RESULTS OF ANN TRAINED BY USING EKF WITH TRANSFORMED INPUT VECTORS.

The network is trained by using EKF with transformed 2-dimensional input vectors. The initial value for $T_{\text{max}}$ of 20 and initial value for the trace of the inverse matrix $Q$ as 20 are used without $\alpha$. The configuration of the network is 2-6-1.

A maximum classification performance of 95.18% is obtained in 8 iterations at MSE of 0.4666, whereas the classification performance of the network is only 89.15% (Figure 2.12) when trained without transforming the original input patterns. Because of the transformation of the input patterns, i) the classification performance has increased; ii) the iterations, at which maximum classification performance is obtained, are reduced, and iii) the size of the network is reduced. The classification performance and the MSE curves are shown in Figure 6.3.

6.5.3 CONVERGENCE CURVES OF ANN TRAINED BY USING BPA AND EKF WITH TRANSFORMED INPUT VECTORS AND PLANAR DISPLAY OF THE TEST PATTERNS

The convergence of EKF is faster than that of BPA. The MSE for BPA starts at a higher value compared to that of EKF. It requires only 7
iterations for EKF and 22 iterations for BPA, to reach MSE of 0.6. The convergence curves for the network trained, by using BPA and EKF with transformed input vectors, are shown in Figure 6.4. The distribution of the test patterns by the network trained, by using BPA with transformed input vectors, are shown in Figure 6.5. The distribution of the test patterns by the network trained, by using EKF with transformed input vectors, is shown in Figure 6.6.

The network trained with transformed input patterns has multifold advantages over that of the network trained, without transforming the input patterns. The advantages are:

i) The number of nodes in the input layer are reduced,

ii) The iterations, at which maximum classification performance is obtained, are reduced, and

iii) There is improvement in the classification performance of the network.

The comparisons of the performances of the network trained with and without transformation of input patterns are given in Table 6.2
Table 6.2  Comparisons of the performances of the network trained
with and without transformed input patterns

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Configuration</th>
<th>Classification performance in percentage</th>
<th>Iterations</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPA</td>
<td>Without transformed input vector</td>
<td>17-6-1</td>
<td>96.36</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>With transformed input vector</td>
<td>2-6-1</td>
<td>96.36</td>
<td>9</td>
</tr>
<tr>
<td>EKF</td>
<td>Without transformed input vector</td>
<td>17-6-1</td>
<td>89.16</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>With transformed input vector</td>
<td>2-6-1</td>
<td>95.18</td>
<td>8</td>
</tr>
</tbody>
</table>
Fig. 6.3 MSE and Classification Performance of the network trained by using EKF with transformed inputs of the renal pattern
Fig. 6.4 Convergence curves of the network trained by using BPA and EKF with transformed input vectors
Fig. 6.5 Distribution of the test patterns for the network trained by using BPA with transformed input vectors
Fig. 6.6 Distribution of the test patterns for the network trained by using EKF with transformed input vectors