CHAPTER 4
IMPLEMENTATION OF BACK PROPAGATION ALGORITHM NEURAL NETWORK FOR STEGANALYSIS

4.1 IMPLEMENTATION OF BPA

The BPA uses the steepest-descent method to reach a global minimum. The number of layers and number of nodes in the hidden layers are decided. It uses all the 3 layers (input, hidden and output). Input layer uses 2 nodes, hidden layer has 2 nodes and the output layer includes two nodes. The architecture of BPA Neural Network is given in Figure 4.1.

Flow-chart for BPA is shown in Figure 4.2. The connections between nodes are initialized with random weights. A pattern from the training set is presented in the input layer of the network and the error at the output layer is calculated. The error is propagated backwards towards the input layer and the weights are updated. This procedure is repeated for all the training patterns. This forms one-iteration. At the end of iteration, test patterns are presented to ANN and the prediction
performance of ANN is evaluated. Further training of ANN is continued till the desired prediction performance is reached.

**Fig. 4.2 Flow-chart of BPA**

**STEPS INVOLVED IN**

**FORWARD PROPAGATION**

The weights of the network are initialized.
The inputs and outputs of a pattern are presented to the network.
The output of each node in the successive layers is calculated.
\[ O(\text{output of a node}) = \frac{1}{1 + \exp(-\sum w_{ij} x_i)} \quad (4.1) \]

The error of a pattern is calculated
\[ E(p) = \frac{1}{2} \sum (d(p) - o(p))^2 \quad (4.2) \]

**REVERSE PROPAGATION**

The error for the nodes in the output layer is calculated
\[ \delta_{\text{output layer}} = o(1-o)(d-o) \quad (4.3) \]

The weights between output layer and hidden layer are updated
\[ W(n+1) = W(n) + \eta \delta_{\text{output layer}} o_{\text{hidden layer}} \quad (4.4) \]

The error for the nodes in the hidden layer is calculated
\[ \delta_{\text{hidden layer}} = o(1-o)\sum \delta_{\text{output layer}} W_{\text{updated weights between hidden & output layer}} \quad (4.5) \]

The weights between hidden and input layer are updated.
\[ W(n+1) = W(n) + \eta \delta_{\text{hidden layer}} o_{\text{input layer}} \quad (4.6) \]

The above steps complete one weight updation.

Second pattern is presented and the above steps are followed for the second weight updation. When all the training patterns are presented, a cycle of iteration or epoch is completed. The errors of all the training patterns are calculated and displayed on the monitor as the MSE.
\[ E(\text{MSE}) = \sum E(p) \quad (4.7) \]

**4.2 RESULTS AND DISCUSSION**

To evaluate the performance of the proposed method, BPA was trained on 1024 images of Group 1 (no hidden message), and 512 images of Group 3 (steganographic images for training). Then, patterns from 256 untrained images (Group 4) were calculated and provided as input to BPA for testing. Some of the Sample cover images and steganographic images are given in Table 4.1.
Table 4.1. Images used for steganalysis by BPA

<table>
<thead>
<tr>
<th>Sample cover images (Group 1)</th>
<th><img src="image1" alt="Sample cover images" /></th>
<th><img src="image2" alt="Sample cover images" /></th>
<th><img src="image3" alt="Sample cover images" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>Steganographic images for training (Group 3)</td>
<td><img src="image4" alt="Steganographic images" /></td>
<td><img src="image5" alt="Steganographic images" /></td>
<td><img src="image6" alt="Steganographic images" /></td>
</tr>
<tr>
<td>Steganographic images for testing (Group 4)</td>
<td><img src="image7" alt="Steganographic images" /></td>
<td><img src="image8" alt="Steganographic images" /></td>
<td><img src="image9" alt="Steganographic images" /></td>
</tr>
</tbody>
</table>

Fig. 4.3 Intensity values of feature-1 of cover image

Fig. 4.4 Intensity values of feature-2 of cover image
Figure 4.3 and Figure 4.4 show intensity values of the patterns used for training ANN. These intensity values correspond to upper nibble of cover images. The maximum intensity value visible is “15” which is equal to “1111”. The number of patterns shown is 7689 in this plot. During training the ANN, two features are used.

Fig. 4.5 The complexity of BPA during training

Figure 4.5 describes the nature of training took by network. From this, it can be observed that the error is drastically reduced from 27 to almost 0 in 2\textsuperscript{nd} iteration, but it still took another 4 iterations to reach required MSE.
Fig. 4.6 Detection of location of message by BPA

In Figure 4.6, “Original” refers to the actual information of the image, and Detected information indicates that the suspect image is a steganographic one. The method produced a positive classification of 94.5% and 5.5% of misclassification. This Figure presents the information detected in ‘✓’. Some of the sample cover images and steganographic images are given in Table 4.1. The information is detected from these steganographic images. ‘✓’ represents the pixels in cover image.

4.3 SUMMARY

This chapter has presented the implementation of BPA neural network for steganalysis. The detection of location of message by using BPA is presented. Chapter 5 presents the implementation of functional update back propagation algorithm for steganalysis.