FAULT DIAGNOSIS USING A DECISION TREE OF SIMPLE MODULAR NEURAL NETWORKS

P. R. Drake & M. K. Kidwai

*Intelligent Systems Laboratory,*

*School of Engineering,*

*University of Wales Cardiff,*

*P. O. Box 688, Newport Road, Cardiff CF2 3TE, UK.*

*Email:* Drake@cf.ac.uk, KidwaiMK@cf.ac.uk

ABSTRACT

A method of fault diagnosis using simple modular neural networks in a decision tree is proposed. The diagnostic accuracy of such a classifier is shown to be better than a single holistic neural network when applied to diagnosing faults in a seven component RC–network.

1 INTRODUCTION

Neural networks have been used widely and effectively for fault diagnosis. Typically the approach is one of using a single holistic neural network with a large number of outputs, one for each fault class. Such a large network can be difficult to train and analyse when the number of fault classes is large. The approach presented here is to use simple modular neural networks – one for each fault class.

Modular neural networks (MNN’s) have the advantages over large holistic networks that they are small and simple, faster to train and
potentially more accurate. Instead of using a single $k$-output neural network to distinguish between $k$ classes, $k$ two-output MNN’s are introduced - one for the detection of the presence or absence of each single fault class. These modular neural networks are formed into a decision tree in which the faults are diagnosed as they are filtered down the tree.

2 SYSTEM UNDER TEST

The system under test (SUT) is the seven component RC circuit in Fig. 1. This circuit was chosen because it has been used extensively elsewhere [1] for testing methods of fault location. It has the following transfer function:

$$\frac{V_0(s)}{V_i(s)} = \frac{a_0 + a_1s}{b_0 + b_1s + b_2s^2 + b_3s^3} \quad (1)$$

Figure 1 – Seven Component RC circuit
where the coefficients of the transfer function are functions of the network elements as follows:

\[
\begin{align*}
a_0 & = R_2 \\
a_1 & = R_1 R_2 C_1 \\
b_0 & = R_1 + R_2 \\
b_1 & = R_1 C_1 R_2 + R_2 R_4 C_3 + (C_2 + C_3)(R_3 R_2 + R_1 R_1 + R_2 R_1) + R_1 R_4 C_3 \\
b_2 & = R_1 R_2 R_3 C_1 (C_2 + C_3) + R_1 R_2 R_4 C_3 (C_1 + C_2) + R_3 R_4 C_2 C_3 (R_1 + R_2) \\
b_3 & = R_1 R_2 R_3 R_4 C_1 C_2 C_3
\end{align*}
\]

For test purposes the circuit elements are assigned the following nominal component values:

\[
\begin{align*}
R_1 & = R_4 = 1\text{M}\Omega \\
R_2 & = 10\text{M}\Omega \\
R_3 & = 2\text{M}\Omega \\
C_1 & = 0.01\mu\text{F} \\
C_2 & = C_3 = 0.001\mu\text{F}
\end{align*}
\]

### 3 FAULT SIMULATION PROCEDURE

The transfer function is used to stimulate the circuit in a computer. Example component fault levels are then drawn randomly from a normal distribution with a mean of 0\% and a standard deviation of 40\% - excluding values within the range \(\pm 40\%\) which are not deemed to be faulty. These fault levels are then applied to the nominal values of the components being simulated as faulty. This simulation includes simulation of the production tolerances in each non-faulty component value by giving each of them a different random perturbation drawn from a normal distribution with a mean of 0\% and a standard deviation of 3\%. Whilst this may seem excessive, it produces a severe test.
4 CIRCUIT TEST PROCEDURE

Frequency response testing is used. For each of the faults simulated the gain of the circuit is recorded at 20, 40, 200 and 400 rad/sec, and the phase at 2000 rad/sec. A previous study [1] reported the suitability of these measurements for testing the circuit.

5 NORMALISATION OF NEURAL NETWORK INPUTS

These measurements are then normalised to remove the effects of different scales and ranges. This is done here by the commonly used method of assuming normality and transforming the values into a standard unit normal distribution using the transformation:

\[ Z = \frac{x - \mu}{3\sigma} \]  \hspace{1cm} (2)

where \( \mu \) is the mean and \( \sigma \) the standard deviation of the original distribution, \( x \) is the original measurement and \( Z \) is a new transformed variable with a standard normal distribution (mean = 0 and standard deviation = 1).

6 NEURAL NETWORK TRAINING

These normalised inputs are then used to train the neural networks. Simple backpropagation is used as the training algorithm. The training set consists of 40 examples of each fault class, including fault free. The test set consists of another 100 examples of each fault class.

7 EXPERIMENT 1: THE HOLISTIC NEURAL NETWORK

Experiments were performed with a four input – eight output (one per fault class) neural network with one hidden layer. The number of neurons in this layer was varied along with the learning rate and momentum to improve the classification accuracy. The best results were
obtained with 12 neurons in the hidden layer, a learning rate of 0.1 and a momentum of 0.1. The results are presented in Table 1.

8 EXPERIMENT 2: REFINED MNN’s

Eight 2-output MNN’s, one for each fault-class, were trained to detect the presence or absence of their assigned class by firing the corresponding output neuron. Then, each MNN was tested using the test set. At this point the MNN for R3 gave the most correctly classified examples so it was chosen as the root of the decision tree.

The test and training examples classified as ‘R3 faulty’ were then deleted from the test and training sets respectively. The remaining ‘filtered’ data was then used to retrain and test the other MNN’s. At this point the MNN for C1 gave the most correct classifications when tested. This was chosen as the next MNN to be added to the decision tree. This process was repeated until the complete decision tree illustrated in Figure 2 had been constructed. Table 1 gives the results of applying the decision tree to the complete test set.

9 EXPERIMENT 3: UNREFINED MNN’s

The MNN’s formed in the previous experiment are ‘refined’ MNN’s since they are retrained as the training data is filtered down the decision tree. In order to examine the effectiveness of this refinement, a decision tree with the same structure as that in Figure 2, but formed of unrefined MNN’s, was constructed and tested. These unrefined MNN’s being those formed at the first stage of the previous experiment with the whole training set. Table 1 gives the results of applying the decision tree of unrefined MNN’s to the complete test set.

10 ANALYSIS OF RESULTS

The decision tree of refined MNN’s gives the most correct classifications. It is important to note that the decision tree of unrefined MNN’s gives virtually the same (marginally worse) overall performance as the single holistic neural network. This implies that breaking a large problem down into small problems is only beneficial, in the context of this application of neural networks, if the refinement process is applied.
### Table 1: Experiment Results

<table>
<thead>
<tr>
<th>Fault Class</th>
<th>Percentage successful diagnosis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Method1</td>
</tr>
<tr>
<td>Fault Free</td>
<td>99</td>
</tr>
<tr>
<td>R1</td>
<td>100</td>
</tr>
<tr>
<td>R2</td>
<td>91</td>
</tr>
<tr>
<td>R3</td>
<td>97</td>
</tr>
<tr>
<td>R4</td>
<td>94</td>
</tr>
<tr>
<td>C1</td>
<td>96</td>
</tr>
<tr>
<td>C2</td>
<td>86</td>
</tr>
<tr>
<td>C3</td>
<td>96</td>
</tr>
<tr>
<td>Average</td>
<td>94.9</td>
</tr>
</tbody>
</table>

**Method 1 = Single Holistic Neural Network**

**Method 2 = Tree of Refined Modular Neural Networks**

**Method 3 = Tree of Unrefined Modular Neural Networks**
Figure 2: Tree of RMNN's
11 ALTERNATIVE DECISION TREES

The application presented here employs a skewed decision tree. It is possible to construct a more balanced tree of MNN’s in which the diagnostic process is one of dividing the faults up into successively smaller groups, until individual faults are diagnosed [2].

12 FURTHER WORK

At the time of writing, this work is being extended to the classification of ECG traces. This particular application involves a large number of classes which require a correspondingly large holistic neural network. Therefore, it is a potentially ideal application for modular neural networks.

13 CONCLUSION

The ability of a single holistic neural network to diagnose faults in a seven component RC-network has been compared with that of a decision tree of simple modular neural networks. It has been found that the decision tree gives more correct classifications, provided that the modular neural networks are progressively refined, by being training on only those training examples that reach their stage of the decision tree.

REFERENCES