Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Sound and Vibration 293 (2006) 96–111
JOURNAL OF SOUND AND VIBRATION
www.elsevier.com/locate/jsvi

A novel information-gap technique to assess reliability of neural network-based damage detection

S.G. Pierce*, K. Worden, G. Manson

Dynamics Research Group, Department of Mechanical Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, UK

Received 1 October 2004; received in revised form 19 September 2005; accepted 22 September 2005
Available online 1 December 2005

Abstract

The application of neural network classifiers to a damage detection problem is discussed within a framework of an interval arithmetic-based information-gap technique. Using this approach the robustness of trained classifiers to uncertainty in their input data was assessed. Conventional network training using a regularised Maximum Likelihood approach is discussed and compared with interval propagation applied as a tool to evaluate the robustness of a particular network. Concepts of interval-based worst-case error and opportunity are introduced to facilitate the analysis. The interval-based approach is further developed into a network selection procedure capable of significant improvements (up to 22%) in the worst-case error performance over a conventional network trained on crisp (single-valued) data.

1. Introduction

The background to the current work lies in the application of artificial neural network (ANN) classifiers to the interpretation of data for the purposes of damage detection. For example, when interpreting low frequency modal vibration data, the discrimination required between complex measured frequency spectra is a typical application where conventional multilayer perceptron (MLP)-type networks may help to facilitate reliable classification. ANN classifiers have found diverse applications in finance, speech, handwriting and facial recognition, product inspection, medical diagnosis and fault detection [1]; it is the last of these applications areas that is studied in the present work. Recent application examples of neural network fault detection include aircraft wing damage detection [2,3], bearing assessment [4], steel frame structure assessment [5] and structural parameter monitoring for nonlinear systems [6,7].

Conventional network training can be viewed within a framework of the global optimisation (minimisation) of the error function between the network output prediction and the true target data. There exist a number of strategies to locate this minimum, the most common being variants of gradient descent such as scaled conjugate gradients [8,9]. These techniques make use of the local gradient information of the error surface to ascertain the optimum search direction but are susceptible to the danger of locking a solution into a local minimum rather than locating the true global minimum. A number of strategies have been devised to counter

*Corresponding author. Tel: +44 0 114 222 7827; fax: +44 0 114 222 7890.
E-mail address: s.g.pierce@sheffield.ac.uk (S.G. Pierce).

0022-460X/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jsv.2005.09.029

this problem [8–10], and new techniques of searching the error function space are currently active areas of research, e.g., using genetic algorithms [4,11].

An additional complication to network performance lies in the capability of a network to generalise its classification performance to previously unseen data. There exists the widely recognised problem of overtraining that can occur such that a network starts to learn the noise present in data rather than the underlying data structure. The uses of cross validation and early stopping [10] using an independent validation data set are often used as termination criteria for training to help avoid this problem. Although the problem is often associated with larger networks, there exists evidence to suggest that generalisation performance depends more on the size of the weights rather than their number [12]. Recent years have seen the development of more sophisticated methods of addressing the overtraining problem through regularisation techniques. The problem with overtrained networks can be understood as a tendency to converge to weight matrix solutions with high component values that tend to generate excessively sharp decision boundaries (where the sharpness is more characteristic of noise in the data) [13]. Regularisation seeks to penalise the large weight and bias values of the network that are associated with sharp decision boundaries. The simplest way to implement such regularisation is using an additive weight decay term in the objective (error) function. A more sophisticated basis for regularisation can be found in the approach of Bayesian network training [8,9,13] which frames the optimisation problem rather differently. The traditional approach (referred to as Maximum Likelihood) seeks to find a unique value of the network weight and bias matrix corresponding to the optimised value. The Bayesian framework marginalises over the weights, assuming that the solution weight matrix has a posterior probability distribution (which will likely be centred close to or at the Maximum Likelihood solution). Such an approach is useful as network weight regularisation falls naturally into the framework; and additionally it is possible to estimate confidence bounds on the output predictions [8,9,13] based on the widths of the posterior probability functions for the weight matrix.

The problems of network overtraining and lack of generalisation are central to understanding the inertia to the practical application of ANNs especially to safety critical applications. If, e.g., we envisage an ANN classifier being used to assess the condition of a major structural airframe component, it is imperative that the performance of the classifier to the most diverse range of inputs is well understood. Poor classifier performance could result in catastrophic failure and possible loss of life. Although a range of techniques (including the Bayesian approach detailed above) have been developed for output confidence interval predications [9,13–16] they all adopt a probabilistic standpoint and therefore suffer from the common drawback that since the probability distributions are usually estimated from the low-order moments of the data (typically mean and standard deviation), there is often no representation associated with the extremes of the distributions. Unfortunately, it is often the extreme events of the data that are likely to be associated with the unpredictable failure events of greatest interest.

The current work comprises a novel non-probabilistic approach applied to predicting extreme network outputs in the presence of uncertainty in the input data. This technique is based on the theory of convex models and information-gap uncertainty as pioneered by Ben-Haim and co-workers [17–20]. We extend interval-based [21] techniques described in previous work on nonlinear regression analysis [22], to investigate the response of a simple MLP network used for a classification problem to locate damage sites on the wing of a Gnat trainer aircraft. The choice for an MLP network architecture was determined as they have been shown [9] "to be universal approximators in the sense that they can approximate to arbitrary accuracy any continuous function from a compact region of input space provided the number of hidden units is sufficiently large and provided the weights and biases are chosen appropriately". However, the techniques described in this paper can easily be extended to accommodate other network architectures if desired.

A comparison with conventional network training based on cross validation is presented (using conventional Maximum Likelihood training incorporating weight regularisation). We show that the use of interval-based network propagation allows a new criterion for network selection to be established. This technique allows the identification of an ANN classifier which is intrinsically more robust to noise on the input data than network solutions obtained by conventional Maximum Likelihood training. Furthermore, by virtue of the conservative nature of interval sets, the reliance on probabilistic-based estimates of confidence bounds on network predictions is obviated. The interval-based worst-case error predictions represent an inclusive bounded solution set given a specified degree of input noise to the classifier.

2. Experimental data collection and processing

2.1. Introduction

Previous work [2,3] has described the application of a structural health monitoring strategy to the problem of damage location on an aircraft (Gnat) wing located at DSTL Farnborough. The wing was instrumented with an array of 12 accelerometers (Fig. 1), to measure the response to forced vibration provided using a Ling electrodynamic shaker attached directly below panel P4 on the lower surface of the wing. The shaker was driven using a white Gaussian noise source. The wing had a series of nine removable panels (P1–P9) which could be removed and replaced to provide a reproducible and reversible representation of changing conditions on the wing structure. In this fashion, damage represented by a local change in stiffness properties could be simulated.

Data were collected from all 12 accelerometers for a variety of undamaged (normal condition with all panels in place) and simulated damage data using a DIFA Scadas 24-channel acquisition system controlled by LMS software running on an HP workstation. Note from Fig. 1 that the accelerometers were positioned in three distinct groupings (A, B and C) across the plate. Rather than record the individual acceleration responses, the experiment was configured to record the ratio of measured accelerations between transducer pairs AR/A1, AR/A2, AR/A3, etc. in such a way that the transmissibilities between transducer pairs formed the base measurement [1,2]. In this fashion, there were a total of nine measurement variables recorded. Fig. 2 illustrates a typical raw transmissibility measurement with 1024 spectral lines recorded with 1 Hz width in the frequency range 1024–2048 Hz.

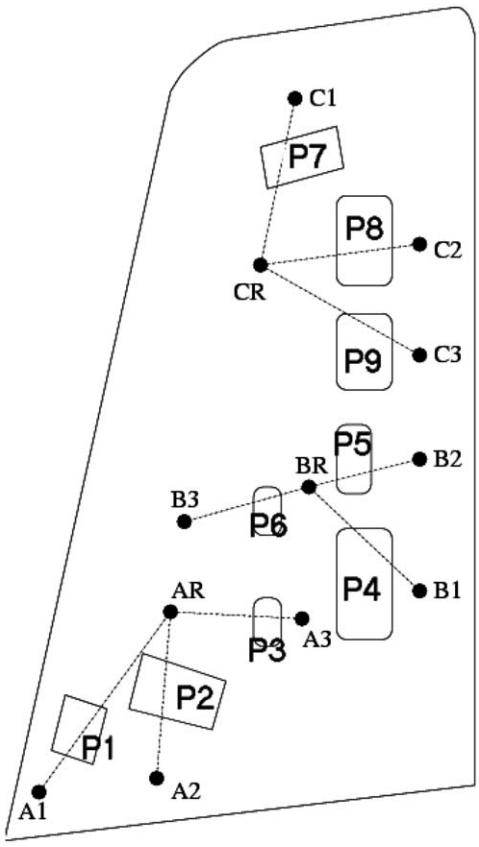

Fig. 1. Detail of Gnat wing showing position of sensors and removable panels.

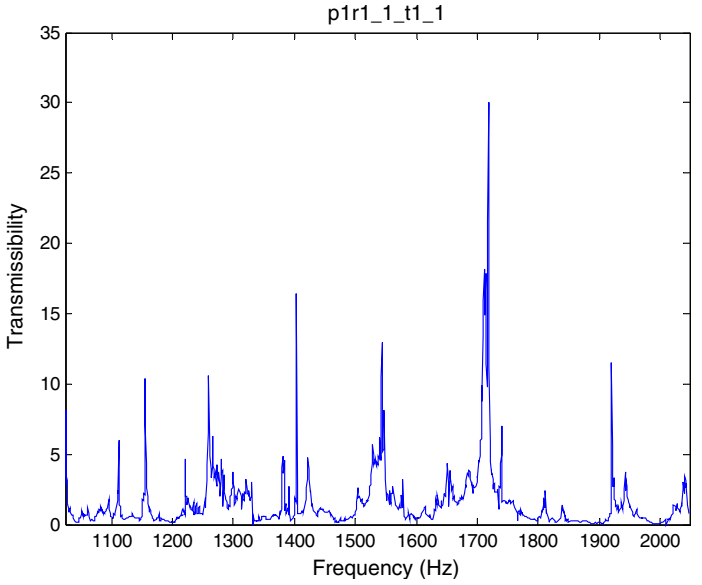

Fig. 2. Example of a transmissibility plot spectrum.

By systematic removal of panels P1–P9, the effect of simulated damage on the spectral response of the transmissibility functions could be observed. For each panel removed, 100 individual measurements were recorded for each of the nine separate transmissibilities (labelled t1–t9) corresponding to removal of panels P1–P9.

In addition to the damage condition measurements, a set of normal condition (undamaged) cases were recorded. Full details of the experimental procedure are given in Ref. [3].

2.2. Data pre-processing

The data were inspected manually to select features (spectral line windows) from the transmissibility spectra (T-spectra) that corresponded to particular damage events. A detailed explanation of this process is documented in Ref. [3]. Essentially, spectral line windows exhibiting the greatest change between ''undamaged'' and ''damaged'' conditions were identified, the problem being considerably simplified by using a non-local argument with respect to the transducer groups [2,3]. For panels P1–P3 only the T-spectra from T1, T2, T3 (corresponding to the accelerometers A1–A4) were considered relevant, and the spectra from transducer groups B and C were ignored. Similarly, for panels P4–P6, only the measurements from transducers B1–B4 were considered. From a total of 77 candidate features [2,3], a single feature was selected (by inspection) to correspond to the removal of an individual panel. In this fashion, the feature set was reduced to nine individual features (F1–F9). For each individual feature the data comprised 700 normal condition measurements, and 1800 test measurements. An outlier (novelty) analysis was then performed [23] by computing the Mahalanobis-squared distance of the data points with respect to the normal condition data:

$$\Delta^2 = (x-\mu)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(x-\mu), \tag{1}$$

where Δ is the Mahalanobis distance, x the data, μ the mean of the normal condition data and $\boldsymbol{\Sigma}$ the covariance matrix of the data.

By performing the novelty analysis for all of the nine features combined a data matrix of size [1800 by 9] was obtained (with each feature corresponding to size [200 by 9]). The data were divided into three equal parts of [594 by 9] to form separate *training*, *validation* and *test* data sets for subsequent network evaluation. (Note since $1800/9/3 = 66.27$, the number of samples was rounded down to 66, thus determining the size of the training, validation and test sets at $66*9 = 594$ rows.) Finally, the data sets were logarithmically compressed and normalised to -1 to $+1$ before presentation to the network.

3. Network topology and training

The MLP network implementation and training was undertaken in MATLAB™ using the NETLAB toolbox developed by Nabney [9]. The data were presented to a series of MLP networks with different numbers of hidden nodes. Each network had nine input nodes corresponding to the features F1–F9, and nine output nodes corresponding to the classes P1–P9 (Fig. 3). The input values were x_i, $i = 1, \ldots, d$. The outputs from the second layer were given by

$$a_k^{(2)} = \sum_{j=1}^{M} w_{kj}^{(2)} \tanh\left[\sum_{i=1}^{d} w_{ji}^{(1)} x_i + b_j^{(1)}\right] + b_k^{(2)},$$
$$j = 1, \ldots, M, \quad k = 1, \ldots, C, \tag{2}$$

where w is the weight matrix, b the bias matrix, M the number of input nodes and C the number of output nodes.

The network output was given by transformation of the second layer activations by the output activation function. Since there were a series of C independent output classes it was appropriate to utilise the *softmax* function [9]:

$$y_k = \frac{\exp\left[a_k^{(2)}\right]}{\sum_{k'} \exp\left[a_{k'}^{(2)}\right]}. \tag{3}$$

The choice of the softmax activation function ensured that the outputs always summed to unity, and thus could be directly interpreted as class conditional probability values. NETLAB automatically used an appropriate entropy function [9] for the 1-of-c output class coding to calculate the network error function and

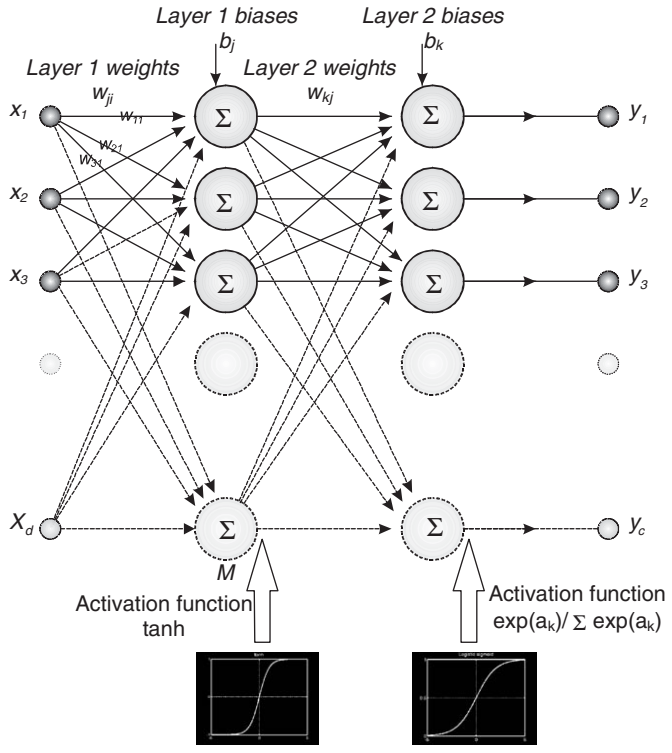

Fig. 3. MLP network structure.

included a weight decay penalty regularisation term:

$$E = -\sum_{n}\sum_{k=1}^{c} t_k^n \ln y_k^n + \alpha \sum w_i^2, \tag{4}$$

where t_k is the target variable, y_k the network prediction and α the weight decay hyperparameter.

The number of hidden nodes in the second layer was varied between 1 and 15 hidden units. Each individual network structure was trained with 100 independent training sessions starting at differently randomly chosen points on the error surface so that a total of 1500 independent networks were evaluated. Up to 1000 iterations of a scaled conjugate gradient optimisation were implemented within a Maximum Likelihood training framework using a small hyperparameter $\alpha = 0.001$ to control weight decay.

4. Conventionally trained network results

Traditional network selection was implemented by dividing the data into three equal portions designated the training, validation and test data sets. The training data were used to train the networks; the best network was then selected from the 1500 possibilities by finding the best classification rate on the validation data set. Finally, the performance of this selected best network was assessed using the test data set. Fig. 4 illustrates the maximum classification rates identified from the validation data set.

First inspection of Fig. 4 would indicate that a network with eight or nine hidden nodes produced an excellent classification performance for a relatively compact network structure. However, if we consider the eight hidden nodes network, this structure has 161 independent components forming the weight and bias matrix. Considering an argument based on the Vapnik–Chervonenkis dimension [8,24], Bishop argues that an approximate rule of thumb to classify correctly a fraction $(1-\varepsilon)$ of new patterns requires a number of training patterns equal to at least

$$N_{\mathrm{MIN}} = W/\varepsilon, \tag{5}$$

where W is the number of components of the weight and bias matrix.

To obtain a classification rate of 90% therefore implies that we require the number of training patterns to be equal to 10 times the number of weights in the network. Since in the current example the total number of training examples was 594 (66 examples of each class), it is clear that the network structure with eight hidden nodes was likely to posses poor generalisation performance. It was for this reason that the current analysis

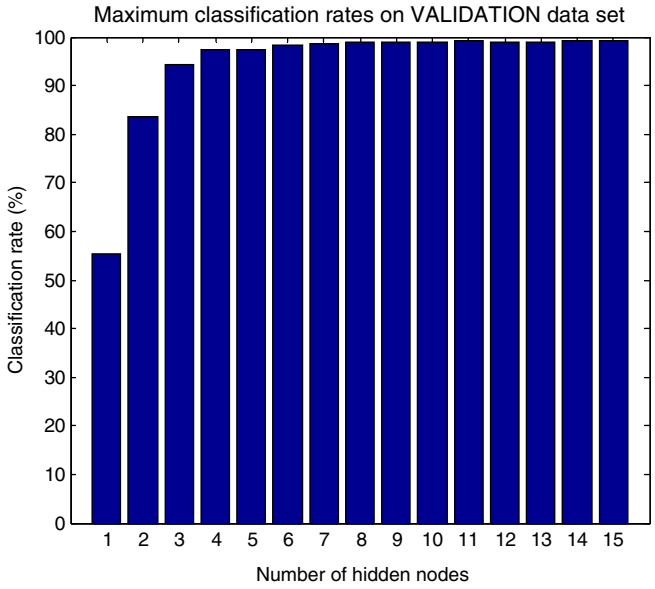

Fig. 4. Maximum classification rates on the validation data set.

Table 1
Maximum and minimum classification rates from training, validation and test data sets (four hidden nodes)

	Training		Validation		Test	
	Min	Max	Min	Max	Min	Max
Classification rate	96.1	100	90.7	97.3	87.7	93.6
Network number	25	4	21	85	48	71

Table 2
Test data confusion matrix, correct classification rate = 92.9%, n_hidd = 4, network #85

Target classes	Network predicted classes								
	C1	C2	C3	C4	C5	C6	C7	C8	C9
C1	61	1	0	0	1	1	2	0	0
C2	2	60	2	0	1	1	0	0	0
C3	1	0	58	3	0	2	0	2	0
C4	0	0	0	63	1	1	0	1	0
C5	1	2	0	2	61	0	0	0	0
C6	0	0	4	1	0	61	0	0	0
C7	0	0	0	0	0	0	66	0	0
C8	0	0	0	1	0	0	3	61	1
C9	0	0	0	0	0	0	2	3	61

restricts discussion to networks with four hidden nodes comprising 22 independent components of the weight and bias matrix. This choice of network provides reasonable classification performance on the validation data, whilst maintaining a relatively simple network structure. Using a cross validation [10] approach to network selection, the best-performing network (with four hidden nodes) from the validation set was selected (found to be network #85), see results in Table 1. Network 85 gave a classification rate of 92.93% when applied to the test data set (see Table 2) showing the classification confusion matrix for the test data. Ideally, the diagonal elements would all be equal to 66 indicating correct classification of all instances of each class. Finally, the variability of all 100 networks for n_hidd = 4 was assessed with respect to the test data. The classification results are illustrated in Fig. 5 also showing the mean and variance of the classification rate for 100 networks. It is clear that acceptably consistent classification performance was obtained for this network structure.

5. Interval-based classification networks and network robustness

5.1. Introduction

Having established the network performance to *crisp* (i.e., single-valued) input data, the next step was to devise a method to investigate the sensitivity of the classification performance of the network to fluctuations in its input data. Quantification of this performance would allow an estimate of the network robustness to be evaluated. Perhaps the most obvious way of performing this analysis would be to use a Monte-Carlo approach, randomising the input data (within certain pre-defined bounds) and monitoring the associated changes in the output classification performance. This technique has a significant drawback especially when applied to nonlinear MLP networks, in that it is impossible to be sure of mapping all possible combinations of variation in input space to output space unless an unfeasibly large number of sample points are used. Since interest lies in understanding the worst possible performance of the classifier in the presence of input data uncertainty, a Monte-Carlo approach would therefore not provide certainty of behaviour under all possible input data conditions. Similarly, the other techniques discussed in the introduction have a similar flaw in that

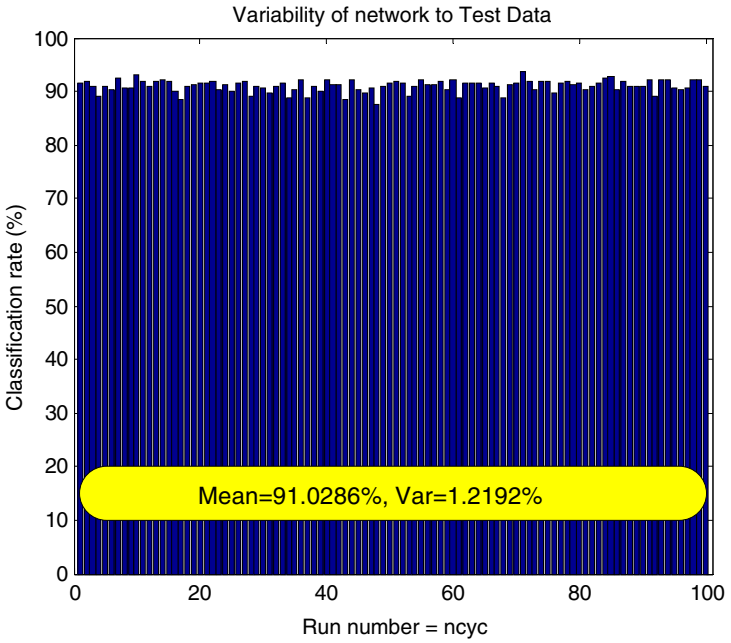

Fig. 5. Variability of classification performance to test data set.

they generate a probabilistic view of the likelihood of the classifier performance with respect to input data fluctuation. It was to circumvent this problem that the input data set was re-defined as a series of interval number inputs. Interval numbers [21] occupy a bounded range of the number line, and can be defined as an ordered pair of real numbers $[a, b]$ with $a<b$ such that

$$[a, b] = \{x | a \leqslant x \leqslant b\}. \tag{6}$$

Interval numbers have specific rules for the standard arithmetic operations of addition, subtraction, multiplication and division [21]:

$$\begin{aligned}
&[a, b] + [c, d] = [a + c, b + d],\\
&[a, b] - [c, d] = [a - d, b - c],\\
&[a, b] \times [c, d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)],\\
&[a, b]/[c, d] = [a, b] \times \left[\frac{1}{d}, \frac{1}{c}\right].
\end{aligned} \tag{7a–d}$$

The MATLAB^{TM} compatible toolbox INTLAB [25] was used to implement the interval calculations required to calculate forward propagation through the MLP networks. This toolbox incorporated a rigorous approach to rounding which was critical when using finite precision calculations in order to preserve the true conservative interval bounds. Formally, with each network, we associated an input set $I(\beta)$ composed of all possible inputs to the network, the size of the uncertainty was governed by the β parameter. Having defined the input set $I(\beta)$, the output response set $R(\beta)$ of all network outputs was computed. It was then possible to quantify the network reliability in terms of how large a β parameter could be tolerated before a point in the failure set (defined by choosing an appropriate threshold for a performance governing parameter, e.g., the worst-case error) was just reached; at this point β attained a critical value β_{CR}. Obviously, a large value of β_{CR} was desirable as the network would then be more robust [20].

Each input value x_i of the test data set was intervalised by a parameter β such that

$$[x_{ia}, x_{ib}] = [(x_i - \beta), (x_i + \beta)]. \tag{8}$$

Obviously, the propagation of interval sets through a crisp valued ANN weight matrix will give rise to interval number outputs. For a regression problem, this is manifest as a set of upper and lower bounds around the true

output prediction [22]. However, for a classification problem, we need to introduce some new definitions for the concepts of the confusion matrix and classification rate to be valid.

5.2. Defining interval classification rates

For crisp valued outputs, the designation for a correct classification (a HIT) is if the class with the highest class conditional probability output (the winning class) belongs to the correct target class. If not it is a misclassification (a MISS). The overall classification rate is just the percentage of HITS to the total number of data presentations. For an intervalised network output with sufficiently small interval bounds we would expect this situation to remain unchanged. The class with the highest bound output remains the winning class (it may be a HIT or a MISS), and its lower bound will be greater than any of the other classes upper bounds. However, as the interval size increases, we will obviously reach a point where one (or more) of the losing class upper bounds becomes equal to or greater than the lower bound of the winning class. (The lower bound of the winning class is defined as the *threshold value* for interval-based classification.) At this critical point, it becomes impossible to distinguish between the two (or more) classes, and either (or any) of the classes could be the winner. If we define the classification HIT rate in terms of a logical OR-type operation between the

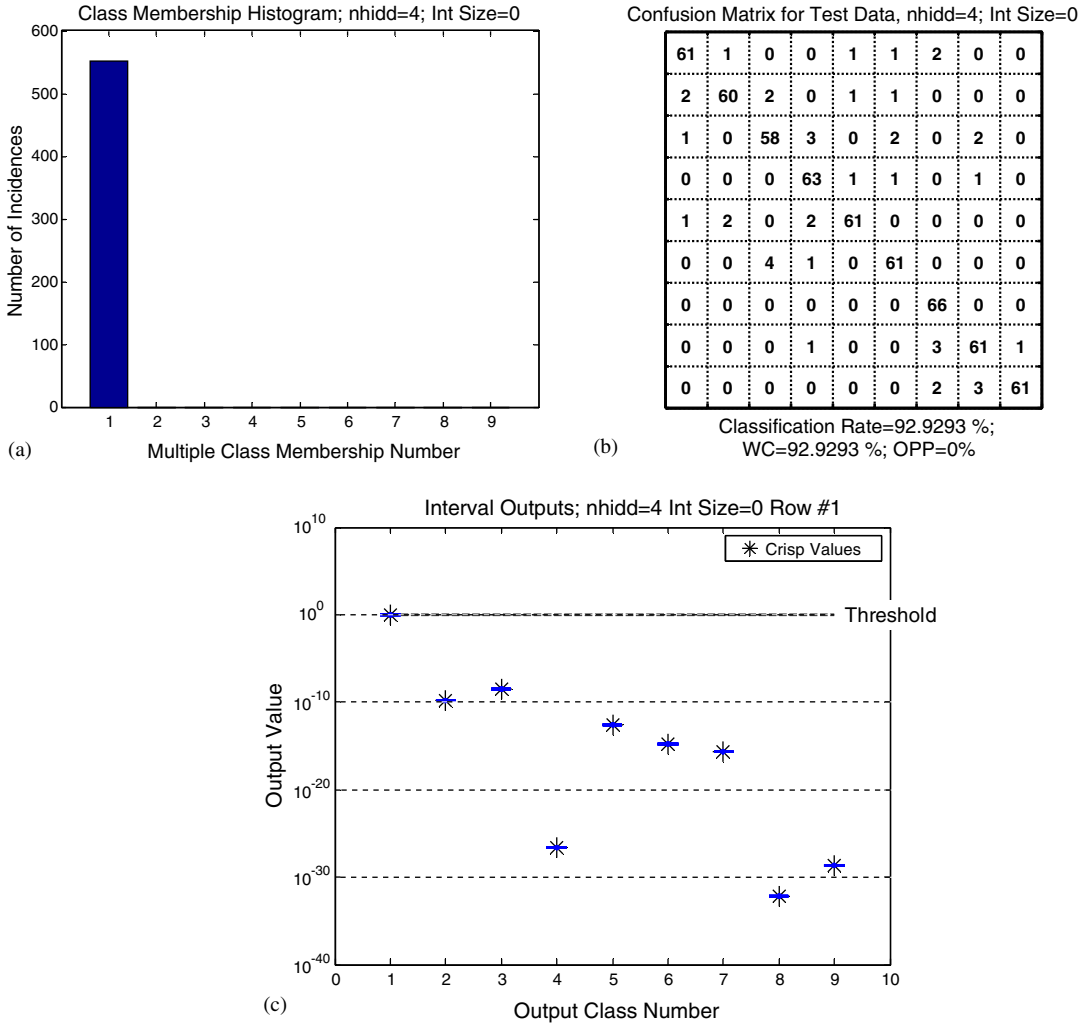

Fig. 6. (a) Class membership histogram, (b) confusion matrix and (c) network output predictions (data vector #1) for interval size $\beta = 0$, * indicates crisp values.

intervalised classes then two effects become important. Firstly, as the interval size is increased the overall classification rate goes up, as it becomes more and more likely that a data vector returns a HIT. However, the possibility of misclassification also increases as more and more of the HITS take on multiple class membership. In keeping with the framework of Ben-Haim [18,20], we define the *opportunity* as the best classification rate minus the crisp classification rate; and the *worst-case* error as the percentage of total number of hits minus the number of hits with multiple class membership.

5.3. Interval network propagation

The concepts of interval classification rate, opportunity and worst-case error are illustrated in the following sequence of figures starting with Fig. 6 showing the results of propagating an interval set with $\beta = 0$ (equivalent to crisp data) through network #85 with four hidden nodes. Fig. 6(a) shows the class membership histogram, note that for the equivalent of the crisp case all of the outputs have single class membership. The confusion matrix (Fig. 6(b)) is identical to the crisp data case in Table 2. The opportunity is 0 and the worst-case error is equal to the best classification rate = 92.9%. Fig. 6(c) shows the actual intervalised network outputs (in this case the intervals are all degenerate, i.e., of zero size).

Fig. 7 illustrates the situation for the same network and input data with a small interval size of $\beta = 0.03$. It is easier to see from Fig. 7(c) how the threshold is defined from the lower bound of the winning class interval.

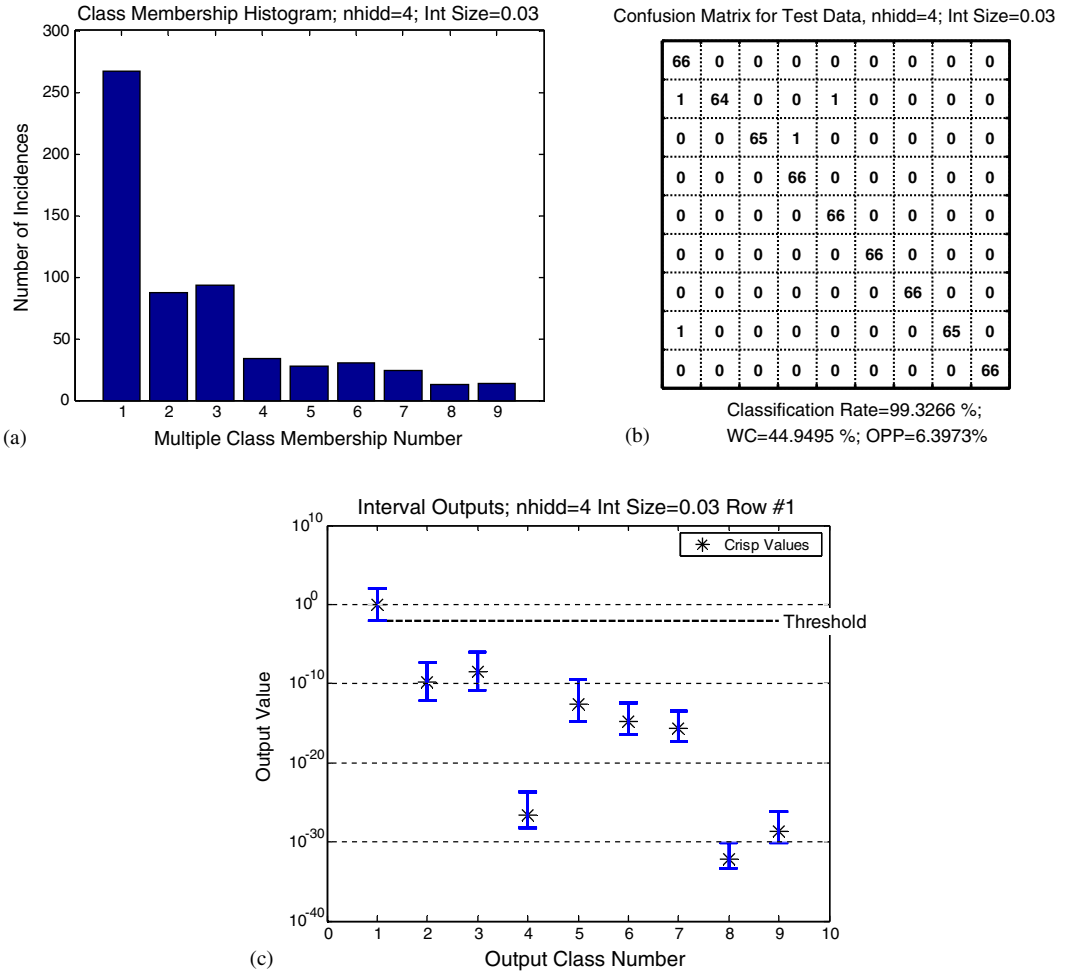

Fig. 7. (a) Class membership histogram, (b) confusion matrix and (c) network output predictions (data vector #1) for interval size $\beta = 0.03$, * indicates crisp values.

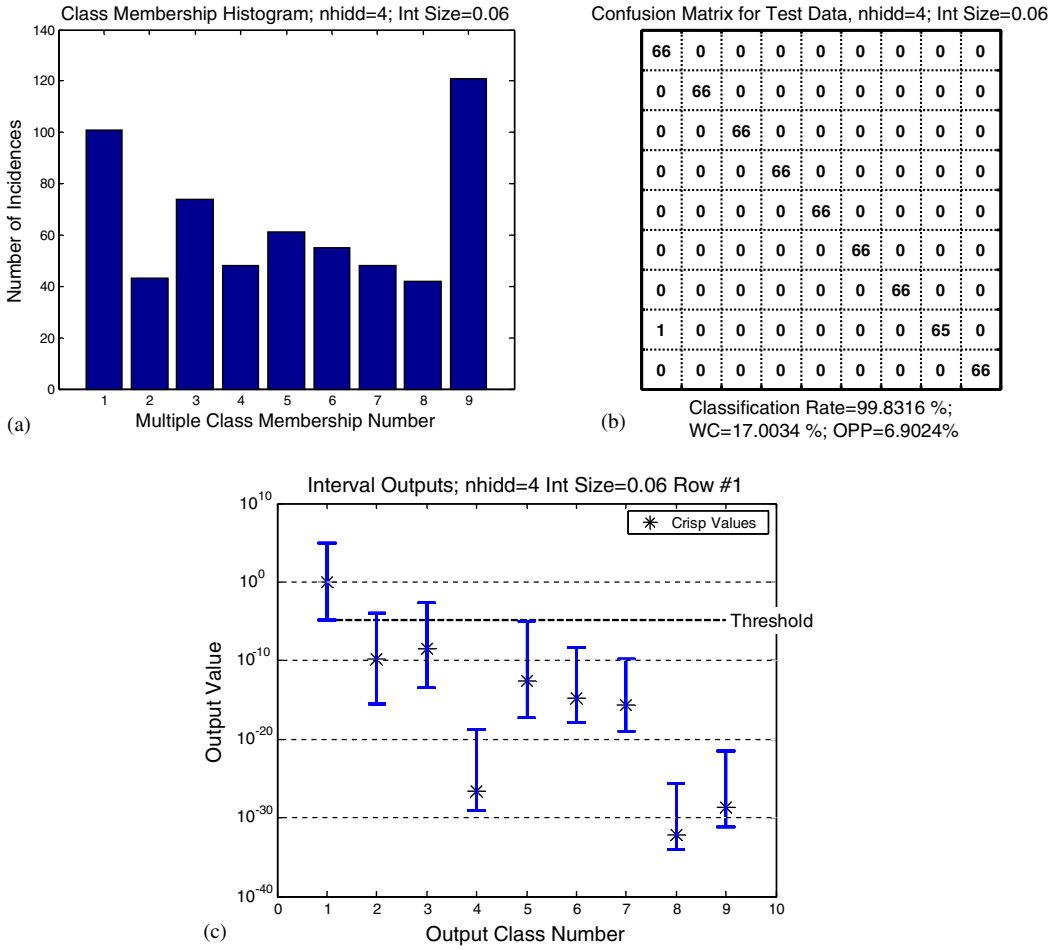

Fig. 8. (a) Class membership histogram, (b) confusion matrix and (c) network output predictions (data vector #1) for interval size $\beta = 0.06$, * indicates crisp values.

For this particular data vector, the classification is still of single membership class; however, it is clear from Fig. 7(a) (the class membership histogram) that looking at the whole test data set, there is significant multiple output class membership. This is reflected in Fig. 7(b) where the best-case classification has risen to 99.3%, with a worst-case error of 44.9% and an opportunity of 6.40%.

Progressing to Fig. 8(c) with an interval size of 0.06, the output intervals for data vector 1 have grown large enough to have multiple output class membership (in this case classes 1, 2 and 3, are all deemed equally likely for calculating the overall classification rate, note that class 5 is close but actually just below threshold in this case). The class membership histogram (Fig. 8(a)) shows a significant increase in data presentations possessing multiple class membership. If the interval size was increased sufficiently we would find eventually that all the data had full nine class membership. This situation would represent complete uncertainty of classification. Finally note that the confusion matrix of Fig. 8(b) indicates a best-case classification of 99.8%, with a worst case of 17.0% and opportunity 6.90%.

5.4. Network robustness

The evolutions of worst-case error, opportunity and best-case classification as a function of input uncertainty parameter (interval size) are illustrated in Fig. 9 which shows the results for a network with four

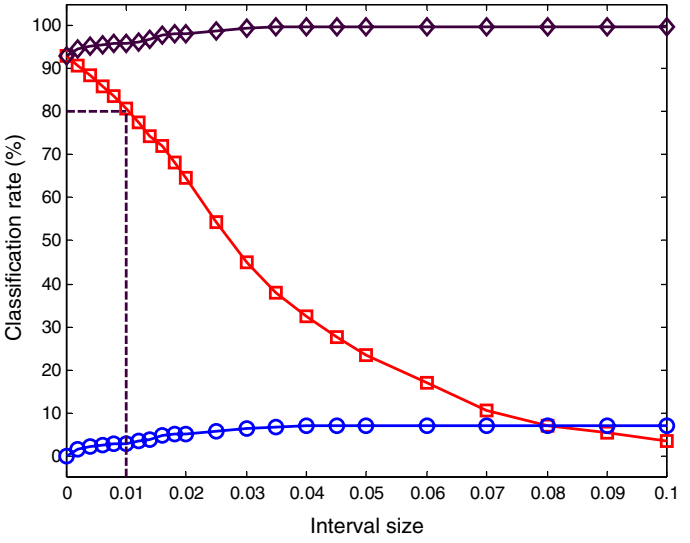

Fig. 9. Classification rate as a function of interval size for network with n_hidd = 4, and network #85, (solid line, square marker) worst-case error, (solid line, circle marker) opportunity, (solid line, diamond marker) best-case error.

hidden nodes. Network #85 was selected as the network structure with the lowest crisp error on the validation set (using the cross validation approach). Fig. 9 is a typical Information-Gap style plot [19,20], where the opportunity is a measure of how much performance headroom is available to the system if we are prepared to tolerate the presence of uncertainty in the input data. However, for safety critical systems, we are more likely to be interested in the worst-case classification rate. From such a figure it is possible to decide on a minimum acceptable classification rate (e.g., 80%) and then infer the corresponding interval size which in turn relates to the spread in the input data. For the case of Fig. 9 the 80% minimum worst-case classification rate corresponds to an uncertainty in the input data of $\beta_{CR} = 0.01$. Since the interval output set is conservative, we can then guarantee that if the input set remains within the bound specified by β_{CR}, the output classification rate cannot fall below the minimum specified rate. In this fashion, the interval forward propagation routine can be used to assess the robustness of an individual network to uncertainty in the input data [26].

6. Interval-based network selection

Having established the basis for the definition of classification rates, worst-case error, opportunity and best-case error applied to interval outputs from an individual network [26]; the technique was extended to investigate propagation through multiple networks to appraise the capability of using interval results as a basis for network selection. The same networks trained using the conventional Maximum Likelihood approach detailed above were investigated. For each particular value of the number of hidden nodes (1–15), the interval forward propagation was evaluated for all 100 individual networks. The range of interval outputs for the worst-case error, opportunity and best-case error were then calculated. The results for four hidden nodes are shown in Fig. 10, the markers indicate the mean values and the error bars show the range (i.e., the maximum and minimum values) of the three functions.

The most interesting feature of Fig. 10 is the large spread in the values of the worst-case error function. For example, at an interval size of 0.02, the worst-case error falls between classification rates of 38.7% and 74.6%. The networks corresponding to these values of the worst-case error set are numbers #46 and #74, respectively (see Table 3). The best network in the sense of tolerance to interval input data was the one that returned the highest classification rate. The results of Table 3 should be compared with Table 1 showing the corresponding network numbers for best classification on the crisp data using Maximum Likelihood training. The best-performing network using conventional cross validation on crisp data was network #85. However, when using

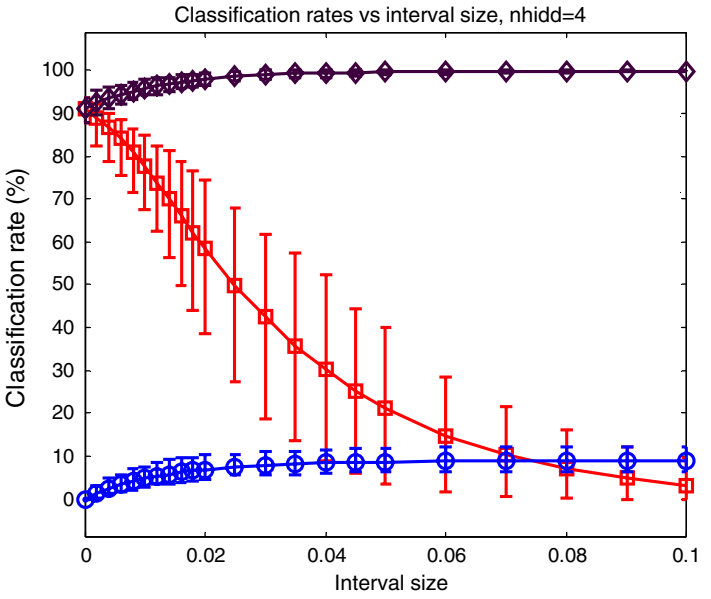

Fig. 10. Interval output classification rates as a function of interval size, (solid line, square marker) worst-case error, (solid line, circle marker) opportunity, (solid line, diamond marker) best-case error.

Table 3
Maximum and minimum interval classification rates as a function of interval size

Interval size	Minimum worst-case error		Maximum worst-case error	
	Value	Network number	Value	Network number
0	87.71	48	93.60	71
0.002	82.49	48	92.26	10
0.004	78.96	48	89.90	93
0.006	75.59	48	88.72	93
0.008	71.38	46	86.36	57
0.010	**67.68**	**46**	**84.85**	**74**
0.012	62.46	80	82.32	57
0.014	56.40	31	81.31	74
0.016	50.00	31	78.79	74
0.018	43.94	31	76.77	93
0.020	**38.72**	**46**	**74.58**	**74**
0.025	27.44	46	68.01	74
0.030	18.86	26	61.78	74
0.035	13.47	26	57.41	93
0.040	8.75	26	52.53	93
0.045	5.89	26	44.44	93
0.050	3.54	26	40.07	93
0.060	1.52	26	28.62	93
0.070	0.51	26	21.72	93
0.080	0.34	26	16.33	49
0.090	0.00	100	12.29	49
0.100	0.00	66	9.76	21

intervalised data, network #85 no longer returned the best classification performance; in fact, the network number that returned the highest (or lowest) classification rate depended on the size of the interval (network #74 for interval sizes 0.01 and 0.02).

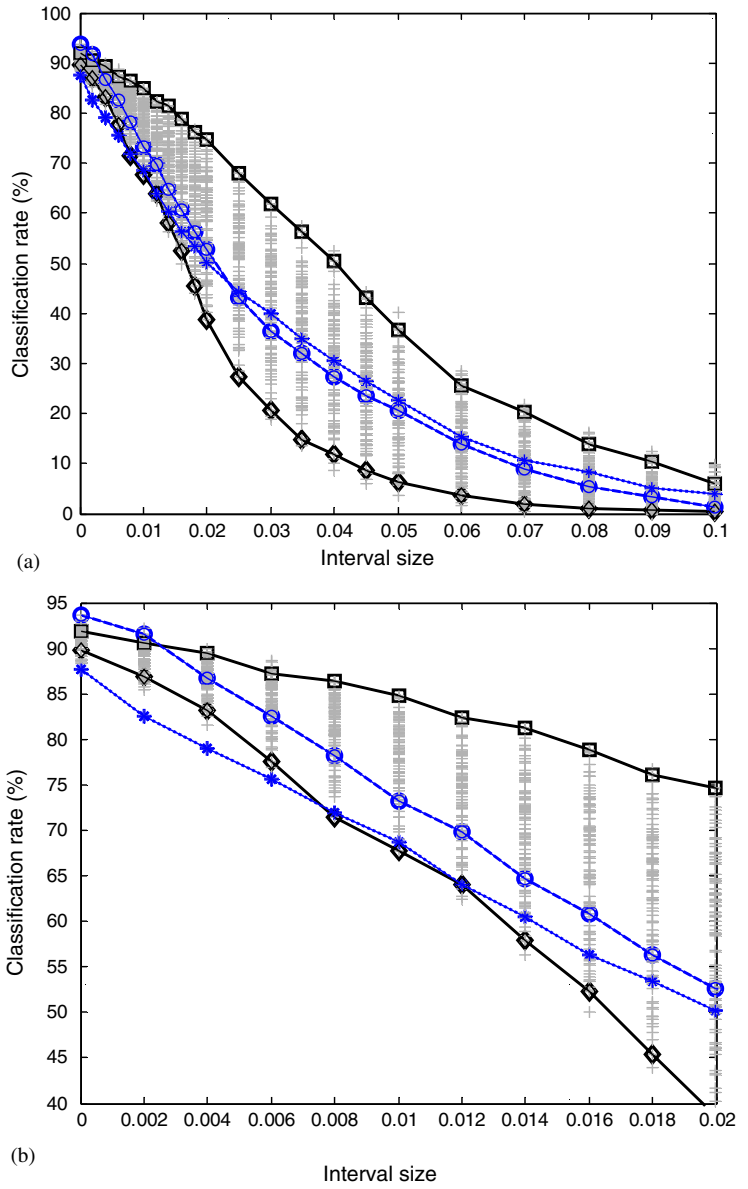

Fig. 11. Comparison between crisp network selection and interval selection for interval size = 0.01 (b) is an enlargement of (a) showing detail for interval sizes up to 0.02, (solid line, square marker) network 74 Interval Data Max (int size = 0.01), (solid line, diamond marker) network 46 Interval Data Min (int size = 0.01), (dashed line, circle marker) network 71 Maximum Likelihood Crisp Data Max, (dotted line, star marker) network 71 Maximum Likelihood Crisp Data Min, (no line, + marker) all Networks.

To illustrate this influence of interval size on choice of network for optimal response to worst-case error, it is instructive to plot the networks giving the highest and lowest classification rates at an interval of size 0.01, compared to the best and worst networks evaluated using crisp Maximum Likelihood on the test data. These results are shown in Fig. 11 where Fig. 11(b) is an enlargement of Fig. 11(a) showing detail for interval sizes up to 0.02.

The grey crosses in Figs. 11(a) and (b) denote all the different interval output results for the 100 networks. It is clear that at an interval size of zero (crisp input data) the Maximum Likelihood results are identical with the interval results evaluated at interval size = 0. (Note, however, they are not identical with the interval results

evaluated at interval size = 0.01.) The best network solution for zero interval size is therefore the conventional Maximum Likelihood result. However, it is clear that for non-zero interval sizes this is not the case. Considering the best interval solution evaluated at an interval size of 0.01, there is an improvement in a worst-case classification rate of 12% over the best Maximum Likelihood solution evaluated on the crisp data. Using the same networks evaluated at an interval size of 0.02 we find that the gain is 22%. However, the best interval solution at interval size = 0.01 is not necessarily the best solution at all interval sizes (although in the current example it is very close). Note that at interval sizes of 0.006, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09 and 0.1 there are grey crosses clearly visible just above the network #74 line indicating a marginally superior network could be used at these interval values.

In general to select the best network in an interval tolerant sense, it was required to set a sensible upper limit on the input uncertainty (the interval size). The best network was then the one giving the highest worst-case error at this uncertainty level. It was imperative to then check to ensure that this network also provided good performance at lower interval values. In this fashion, when considering interval input data, it is possible to determine a more appropriate choice of network selection criteria than is available using conventional Maximum Likelihood training applied to crisp input data.

7. Conclusions

This paper has discussed the application of multilayer perceptron neural networks to a classification problem involving damage detection performed on a GNAT wing. Low frequency (1–2 kHz) modal vibration spectra were recorded for nine different conditions corresponding to local changes in the stiffness matrix of the wing. A novelty analysis was used to perform feature extraction in the frequency domain that corresponded to the individual damage conditions. A series of MLP neural networks were then trained on the data in an attempt to produce a reliable classifier that could be used to predict the location of damage on a previously unseen test data set. Using a conventional Maximum Likelihood training approach based on scaled conjugate gradient descent incorporating weight regularisation, a number of different network structures were evaluated for classification performance. Within a conventional framework of cross validation, candidate network structures were identified and their performance to unseen test data was established to be satisfactory (91.0% mean classification rate over 100 independent networks).

Further analysis involved the forward propagation of interval sets through the trained network structures. The merit of the interval-based forward propagation technique lies in two distinct areas. Firstly, it allows the definition of a critical size of uncertainty in the input data set that guarantees that the output classification performance will not fall below a pre-defined level. This is useful to apply to a single network structure (which could be obtained from any general training technique) to evaluate the robustness of that particular network to uncertainty or noise in the input data. Secondly, and more importantly, an extension of the first technique can be used as an alternative criterion for network selection. We have demonstrated that the worst-case interval output classification rates from conventional MLP networks can be used to find network structures with significantly improved classification performances over their conventional crisp output counterparts in the presence of input uncertainty or noise. In general, it seems that prior knowledge of the maximum size of the input uncertainty is required to select the optimum network structure. By way of example for an interval size of 0.02 (corresponding to 2% noise on the input data) an improvement over conventional training of 22% was possible using the interval approach.

For future work we propose that the current technique be extended to consider multiple location damage events where the cross sensitivity of changes to the transmissibility spectra must also be considered. Additionally, it would be beneficial to directly incorporate an interval tolerance penalty term into the objective function so that it would be possible to automatically select the most robust network to uncertainty of the input data given an additional input parameter of maximum size of uncertainty. It is anticipated that better quantification of the robustness of neural network classifiers will help to promote greater understanding and acceptance of such classifiers in increasingly critical application areas of damage detection.

Acknowledgements

This work was supported by EPSRC grant number RA 013700 in association with DSTL Farnborough who are acknowledged for sample provision and assistance with data collection. The authors acknowledge the use of the software package NETLAB developed by Ian Nabney of Aston University [http://www.ncrg.aston.ac.uk/netlab/ and INTLAB developed by Siegfried M. Rump at Technical University Hamburg-Harburg, Germany [http://www.ti3.tu-harburg.de/~rump/intlab/].

References

[1] G.P. Zhang, Neural networks for classification: a survey, *IEEE Transactions on Systems, Man and Cybernetics—Part C: Applications and Reviews* 30 (4) (2000) 451–462.
[2] G. Manson, K. Worden, D. Allman, Experimental validation of a structural health monitoring methodology. Part II. Novelty detection on a Gnat aircraft, *Journal of Sound and Vibration* 259 (2) (2003) 345–363.
[3] G. Manson, K. Worden, D. Allman, Experimental validation of a structural health monitoring methodology. Part III. Damage location on an aircraft wing, *Journal of Sound and Vibration* 259 (2) (2003) 365–385.
[4] L.B. Jack, A.K. Nandi, Fault detection using support vector machines and artificial neural networks, augmented by genetic algorithms, *Mechanical Systems and Signal Processing* 16 (2–3) (2002) 373–390.
[5] J.L. Zapico, K. Worden, F.J. Molina, Vibration-based damage assessment in steel frames using neural networks, *Smart Materials & Structures* 10 (3) (2001) 553–559.
[6] S.F. Masri, M. Nakamura, A.G. Chassiakos, T.K. Caughey, Neural network approach to detection of changes in structural parameters, *Journal of Engineering Mechanics-Asce* 122 (4) (1996) 350–360.
[7] S.F. Masri, A.W. Smyth, A.G. Chassiakos, T.K. Caughey, N.F. Hunter, Application of neural networks for detection of changes in nonlinear systems, *Journal of Engineering Mechanics-Asce* 126 (7) (2000) 666–676.
[8] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, London, 1995.
[9] I.T. Nabney, *Netlab-Algorithms for Pattern Recognition*, Springer, Berlin, 2002.
[10] S. Haykin, *Neural Networks, a Comprehensive Foundation*, second ed., Prentice-Hall, Englewood Cliffs, NJ, 1999.
[11] D.E. Goldberg, *Genetic Algorithms in Search, Optimisation and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
[12] P.L. Bartlett, The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network, *IEEE Transactions on Information Theory* 44 (2) (1998) 525–536.
[13] D.J.C. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, Cambridge, 2003.
[14] D.J.C. MacKay, The evidence framework applied to classification networks, *Neural Computation* 4 (5) (1992) 720–736.
[15] G. Papadopoulos, P.J. Edwards, Confidence estimation methods for neural networks: a practical comparison, *IEEE Transactions on Neural Networks* 12 (6) (2001) 1278–1287.
[16] D. Lowe, C. Zapart, Point-wise confidence interval estimation by neural networks: a comparative study based on automotive engine calibration, *Neural Computing & Applications* 8 (1999) 77–85.
[17] Y. Ben-Haim, I. Elishakoff, *Convex Models of Uncertainty in Applied Mechanics*, Elsevier, Amsterdam, 1990.
[18] Y. Ben-Haim, *Robust Reliability in the Mechanical Sciences*, Springer, Berlin, 1996.
[19] F.M. Hemez, Y. Ben-Haim, S. Cogan, Information gap robustness for the test-analysis correlation of a nonlinear transient simulation, in: *Proceedings of the Ninth AIAA/ISSMO Symposium on Multi-disciplinary Analysis and Optimisation*, September 4–6, 2002, Grand Hyatt, Atlanta, GA.
[20] Y. Ben-Haim, *Information-Gap Decision Theory*, Academic Press, New York, 2001.
[21] R.M. Moore, *Interval Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1966.
[22] S.G. Pierce, K. Worden, G. Manson, *Information-Gap Robustness of a Neural Network Regression Model IMAC XXII*, January 26–29, 2004, Dearborn, MI, USA.
[23] K. Worden, G.R. Tomlinson, *Nonlinearity in Structural Dynamics*, Institute of Physics Publishing, 2001.
[24] E.B. Baum, D. Haussler, What size net gives valid generalisation?, *Neural Computation* 1 (1) (1989) 151–160.
[25] S.M. Rump, INTLAB-INTerval LABoratory, in: T. Csendes (Ed.), *Developments in Reliable Computing*, Kluwer Academic Publishers, Dordrecht, 1999, pp. 77–104.
[26] S.G. Pierce, K. Worden, G. Manson, Information-gap analysis of a neural network damage locator, *IMechE Meeting on Pattern Recognition-Detection, Classification and Monitoring*, University of Liverpool, July 13, 2004, Liverpool, UK.