CONF-9003200--1

PNL-SA--18095

DE90 017239

CLASSIFICATION OF ACOUSTIC EMISSION WAVEFORMS FOR NONDESTRUCTIVE EVALUATION USING NEURAL NETWORKS

R. S. Barga
R. B. Melton
M. A. Friesel

April 1990

Presented at the
Society of Photo-Optical Intstitution
Engineering Applications of Neural Networks
Orlando, Florida
March 16-19, 1990

Work supported by
the U. S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Richland, Washington 99352

Classification of acoustic emission waveforms for nondestructive evaluation using neural networks

Roger S. Barga
Mark A. Friesel
Ronald B. Melton
Pacific Northwest Laboratory[a]
P.O. Box 999, Richland, WA 99352

ABSTRACT

Neural networks were applied to the classification of two types of acoustic emission (AE) events, crack growth and fretting, from a simulated airframe joint specimen. Signals were obtained from four sensors at different locations on the test specimen. Multilayered neural networks were trained to classify the signals using the error backpropagation learning algorithm, enabling AE events arising from crack growth to be distinguished from those caused by fretting. In this paper we evaluate the neural network classification performance for sensor location dependent and sensor location independent training and testing sets. Further, we present a new training strategy which significantly reduces the time required to learn large training sets using the error backpropagation learning algorithm, and improves the generalization performance of the network.

1. INTRODUCTION

An acoustic emission (AE) is the transient wave resulting from the sudden release of stored energy during a deformation and failure process such as fretting or flaw growth in a material. This energy can be detected by a surface-mounted sensor, which is sensitive to displacement or velocity.

The detection of flaw growth is of particular engineering interest because of the need to nondestructively evaluate the structural integrity of a component *in situ* and to determine if the component has degraded beyond tolerance[1]. Pattern recognition and signal processing techniques have been successfully applied to develop classifiers for detecting flaw growth AE signals[1,2]. While these techniques can produce an accurate classifier, the process of selecting features is often uncertain, and the procedure is time consuming. Moreover, the resulting classifier is often computationally complex to the point of prohibiting real-time processing of AE data for on-line monitoring applications.

In recent years, neural networks have offered an alternative approach to pattern recognition and signal processing based on automated learning procedures for massively parallel networks of simple processing elements (for an excellent review, see Lippman[3]). Neural networks are attractive for detecting flaw growth signals not only because they may provide faster responses but because they are capable of automatically discovering features and *patterns* of interrelated features which serve to define the corresponding class of an AE signal. Further, as a consequence of the learning procedure, which attempts to discover regularities in the training patterns, the neural classifier is capable of generalizing to novel AE signals. Automatic feature discovery and generalization capabilities are essential for a classifier which must process variable information such as AE returns, where the signals vary due to sensor location and changes in the material or environment.

The problem addressed in this paper focuses on the discrimination of waveforms from two different AE returns in a simulated airframe joint specimen. The neural network is presented with waveforms of AE returns from crack and fretting events, and must learn to discriminate each by

---

[a]Operated for the U.S. Department of Energy by Battelle Memorial Institute under contract DE-AC06-76RLO 1830.

discovering features in the waveform. In this paper we evaluate the classification performance for sensor location dependent and sensor location independent training and testing sets. We also present a training strategy which reduces the time required to train the neural network and improves generalization performance.

The following section describes the neural network model, network design and the learning algorithm used in the study. The collection and preprocessing of the acoustic emission returns is described in the third section. The fourth and fifth sections describe the classification experiments and present experimental results. The sixth section discusses strategies for learning and improving the network generalization. The paper concludes with a brief discussion of the results.

2. NETWORK MODEL DESCRIPTION

To be useful for AE signal classification, a neural network must have a number of properties. First, it should have multiple layers of processing elements and sufficient interconnections between the processing elements in each of the layers. This is to ensure that the network will have the ability to learn complex nonlinear decision surfaces[3,4,5]. Second, the actual features or patterns of interrelated features learned by the neural network should be invariant under translation in time. Without this *shift invariance* the neural network would require precise alignment of the AE return. Since this is not always possible, in practice, the network must be able to accommodate slight misalignments of features in the acoustic emission waveform. Third, the number of connection weights in the neural network should be sufficiently small compared to the amount of training data so that the network is forced to encode the data by extracting regularity[5]. If the network has too many connection weights or there are too few training examples, it can "memorize" all of the correct responses. If the number of training examples is large, the learning procedure is forced to discover features common amongst all the examples in the training set; this enables the network to generalize and correctly classify AE returns that were not included in the training set.

In the following sections, we give a brief introduction to the Multilayered Perceptron (MLP) neural network model, that satisfies all of these criteria, and to the backpropagation learning procedure used in this study. Further details of the MLP and backpropagation learning algorithm can be found in Rumelhart *et al.*[5] and in tutorial format in Lippman[3].

2.1 The multilayered perceptron network for pattern recognition

The MLP is a layered network of homogeneous feed-forward processing elements which receive and transmit analog signals. The output signal from an individual element generates input signals for certain other elements after passing through a set of weighted connections. Each of these weighted connections multiplies the transmitting element's output signal by a specific connection strength (weight) and presents that product as an input for the receiving processor. The weight for a given connection can take on any real value, so the effect of a specific element's output may vary considerably from one element to the next. Processing elements in the MLP network generally receive inputs from all processors in the preceding layer of elements and respond to the total received signal. For a given processor, different patterns of output signals at the preceding layer of processing elements will produce varying levels of total input. This behavior arises, because the specific pattern of connection weights will amplify some of the individual source signals more than others. The total input will be particularly high when the source processing elements send strong signals along connections with large weights, and it will decrease as strong signals are shifted to paths with smaller connection weights or the connections with large weights carry smaller signals. Thus, communication through the weighted connections enables the processors to detect differences in the pattern of transmitted signals. An analog transfer function, most commonly a sigmoid function[5], produces the corresponding variations in the output signal. A schematic diagram of a MLP processing element is shown in Fig. 1.

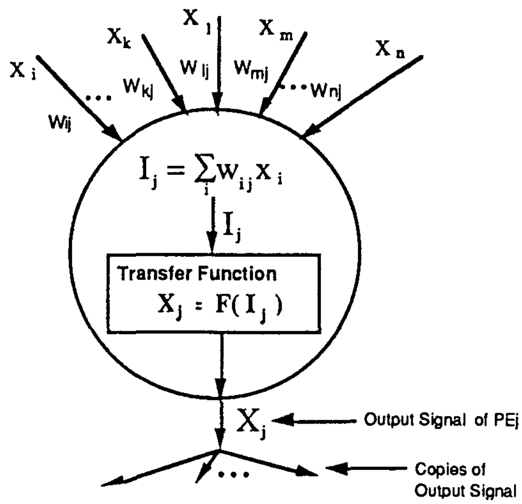

Figure 1 - Model processing element in the MLP network. Weighted connections, W_{ij}, multiply the transmitting processing element's output signal, X_i, and present that product as an input to the receiving processing element. A processing element generates its output signal by applying an analog transfer function to the sum of its input signals.

The MLP's processing elements are organized into distinct layers. All of these layers share a common structure wherein the processors have input connections only from elements in the previous layer and have output connections only to elements in the following layer, and signals traverse these layers in the same order. A schematic representation of this architecture is shown in Fig. 2, where an input pattern enters at the bottom and signals flow up to the top of the network. Output from any given level serves as input for the next layer, until a layer representing the final classification categories is reached. The system is strictly a feed-forward network where signals originate at the initial input layer and propagate towards the final output layer. A hierarchial structure is produced by connecting the processing elements in a "fan in" pattern, so that the number of elements gradually decreases as signals propagate into deeper levels of the system. Under this connection scheme, each processing element receives input signals from elements in the layers which immediately precede it. However, the number of indirect connections between a processing element and more distant predecessors grows significantly as the number of intervening layers increases. For any particular processing element, the complete set of input sources in an earlier layer will be referred to as the elements's "receptive field" on that layer. Since processing elements at deeper levels gain access to progressively larger portions of the input patterns, they can respond to progressively more complicated features, and simpler features will be detected over a progressively larger receptive field. The final output layer consists of elements whose receptive field covers the entire input layer. This hierarchial structure contributes to the MLP's ability to discover interrelated features and extract structure from input data, as well as it's capacity for shift invariant pattern recognition.

2.2 Network design

For the discrimination of acoustic emission signals, we constructed a four layer MLP network. The input layer of the network contained 250 processing elements, with the input level of each clamped to an amplitude value of the waveform to be classified. The input layer can be considered a "window" which looks at the waveform by sampling 250 equally spaced points. The output layer of the network consisted of two processing elements. The output signal of the processing elements in the final layer determined the class of the waveform: (1,0) represented a return from a crack AE event, and (0,1) represented a return from a fretting AE event. Two middle, or *hidden*, layers of processing elements were used in the network since we believed that highly complex decision surfaces were necessary to properly perform classification in the light of the considerable acoustic variability in the returns[b]. The number of processing elements within each hidden layer was selected empirically from a number of alternatives. A diagram of the basic network is shown in Fig. 2.

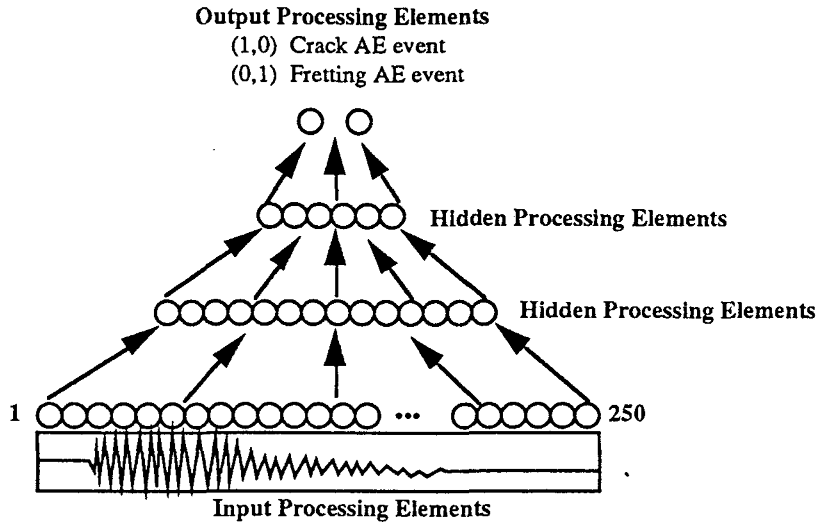

Figure 2 - Architecture of the MLP network. The input layer consists of 250 processing elements with their input signals "clamped" to the amplitude of the AE signal. Two processing elements in the output layer represented the class of the AE signal being processed. The two hidden layers of the network allow the network to extract the high-order signal features required for accurate classification and generalization to novel signals.

2.3 Network training procedures

Several techniques exist for learning in MLP networks[3,5]. For the present study, the error-backpropagation, or simply backpropagation learning algorithm[5] was used. The connection weights in the MLP network were initialized to small random values uniformly distributed in the range [-0.3, 0.3]. This is necessary to prevent the processing elements in the hidden layers from acquiring identical connection weights during training[5]. The classification performance of the network was then gradually improved by changing the connection weights according to the backpropagation learning procedure. In our simulations, the error measured at each processing element at the output layer was back-propagated only when the difference between the actual and desired value of the output processing element was greater than a margin of 0.2. In all experiments, a learning rate of 0.05 and a momentum[5] of 0.3 was used. These learning values are uncommonly small, compared to what we have used in other studies[6], because we found that in order for the learning algorithm to converge, when using large training sets, it was necessary to take very small learning steps.

---

[b]It has been shown in Longstaff[4] and Lippman[3] that two hidden layers of processing elements are sufficient to form any arbitrarily complex decision surface.

The backpropagation learning procedure is rather computationally expensive, due to the many iterations necessary for learning in large networks and the number of examples in the training set. In our case, the training set consisted of approximately 500 AE returns, and between 200 and 3000 epochs (presentations of the whole training set) were required to train the network. Two steps were taken to perform learning within a reasonable time. First, we implemented a distributed simulator for research on large-scale neural networks[7], which allows local graphics workstations to utilize neural network simulators implemented on remote supercomputers and interactively display the results of the simulation. For this study, the MLP network and the backpropagation learning procedure were implemented in vectorized FORTRAN on a Cray X-MP supercomputer, and a graphics interface was implemented in C on a SUN 4 workstation. Our present system achieves a factor of 175 speedup in clock time over a VAX 11/780. The second step taken toward improving the learning time was to develop strategies for training a multilayered neural network, given a large number of training samples to be learned. These strategies are described in section 6. It is important to note, however, that the amount of computation considered here is necessary *only for training* the MLP network classifier and *not for classification*. Classification can easily be performed in real time on a workstation or personal computer.

3. ACOUSTIC DATA COLLECTION

The data used for the network experiments were acoustic returns collected from a test specimen made from 7075-T651 aluminum with a pinned joint to allow production of fretting noise as well as crack growth AE. The specimen was configured such that it generated AE from pin hole fretting and crack growth. Returns were collected from four wideband piezoelectric transducers which were epoxy-mounted to the specimen around the test joint in the positions shown in Fig. 3. The sensors were similar in construction, with a peak response band between 0.2 and 0.7 MHz. The data acquisition instrument used a Biomation 1010 transient recorder with a sample rate of 5 MHz and a buffer size of 4096 points, and a Kennedy recorder. During data collection, waveforms were accumulated and written to tape when an arriving AE event triggered the data acquisition device. Details of the test specimen and the data acquisition instrument are described in greater detail in another publication[1].

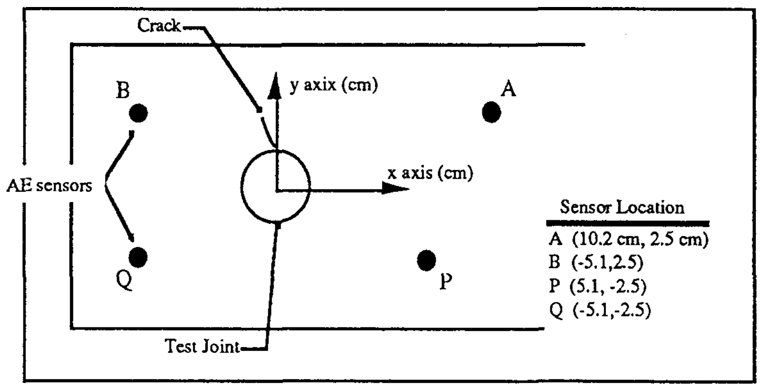

Figure 3 - Sensor locations on specimen.

Data collection from the test specimen was performed in two phases. The first phase was to collect returns from fretting AE events and the second to obtain returns from the crack growth AE events. During the fretting phase, events were recorded that were caused by friction between the pin at the unlubricated test joint and the inside surface of the joint hole. Load was applied to the specimen in the form of a sinusoidal load cycle ranging from 3 to 30 kip. Following the fretting phase, the pin was removed and the specimen was prepared for the second phase of data collection (crack growth phase) by polishing and lubricating the test joint and sawing a 0.64 cm long by 1.0

mm wide vertical notch in the edge of the test hole. After reassembly, cycling was continued with shutdowns every 1.3 mm of crack growth for examination and relubrication of the test joint and for calibration.

A set of 467 returns (230 fretting waveforms and 237 crack growth AE waveforms) were collected. The set of waveforms were sorted by AE event type and sensor location. For each waveform, noise preceding the wavefront was removed and the first 250 microseconds of the waveform was digitized to represent the signal on the input layer of the neural network[c]. Fig. 4 shows a sample waveform from the fretting and crack growth AE event.

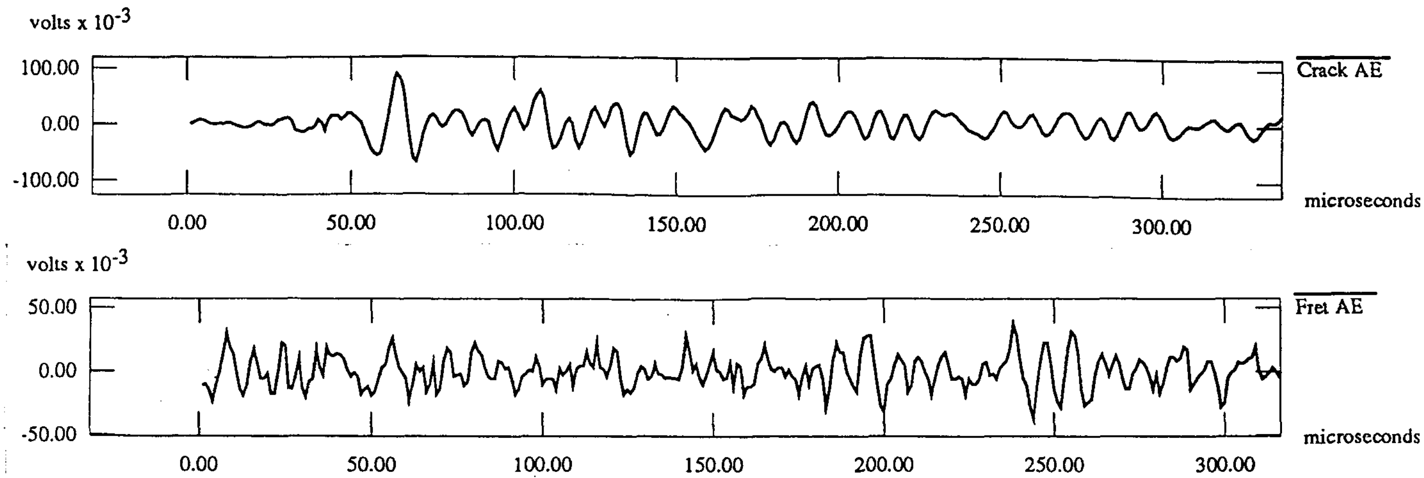

Figure 4 - Examples of crack event and fretting event AE returns from sensor location A.

4. CLASSIFICATION EXPERIMENTS

Classification experiments were designed to address two issues. The first, was to determine whether the MLP network could be trained to discriminate crack growth from fretting AE events. The second, was to evaluate the impact of sensor location information on the classification accuracy of the network. To perform this evaluation, we compare the performance of a network trained using AE returns from each of the four sensors (sensor location dependent) with the performance of a network trained on randomly selected AE returns (sensor location independent) using a test set designed to contain AE returns from all sensor locations. To illustrate the variation in AE returns due to sensor location, Fig. 5 displays the waveforms from each of the four sensor locations from a crack growth AE.

In each experiment, a given network was presented with a sequence of AE returns from the training set and the backpropagation learning algorithm was allowed to adjust the connection weights to improve the classification performance of the network. Training proceeded until the network had learned to correctly classify all of the AE returns present in the training set. During the training process, it was stipulated that the correct response for an AE return of a given class (crack or fretting) should be greater than 0.8 for the processing element corresponding to the returns's class, and less than 0.2 for the other processing element. After training, the network was presented with a set of AE returns excluded from the training set, referred to as the test set, to determine its ability to generalize. For testing purposes, the network was considered to have correctly classified the AE return if the processing element with the greater output signal was associated with the return's class. The network's performance on the test data was specified as the percent correct classification.

---

[c]Naturally, a number of alternative signal representations could be used as input to the neural network, but have not been tried in this study. The digitized waveform was selected for our initial studies because it required no complex feature extraction procedure before classification.

4.1 Single sensor experiment

For this experiment, the training and test sets were created using AE returns from a single sensor location. From the 112 available AE returns for crack and fretting events, seven disjoint test sets consisting of 17 AE returns were selected. The 95 AE returns remaining after each test set selection served as the corresponding training set for the network. In this way, each AE signal in the total set of 12 returns for sensor A served as both a training and test pattern at some point in the experiment. The MLP network was trained and tested on each of the seven training/test set pairs and classification performance was averaged to compute a realistic error rate estimate. This *series* approach, an extension of the "leave one out[8]" method, is designed to accurately compute the error rate estimate of a classifier when a large training set is in use[9].

4.2 Sensor location dependent experiment

For this series of experiments, the training and test sets were designed to ensure that both contained AE returns from each sensor location with equal frequency of occurrence. From the 467 available AE returns, seven training sets were built consisting of 150 AE returns. Seven testing sets were built consisting of 23 AE returns selected from all four sensor locations. These testing sets would be used in evaluating both the sensor location dependent and the sensor location independent network performance.

4.3 Sensor location independent experiment

For the sensor location independent experiment, seven training sets were built consisting of 95 randomly selected AE returns, without regard to the sensor location. The test set used in this series of experiments were the same as the sensor location dependent experiments.

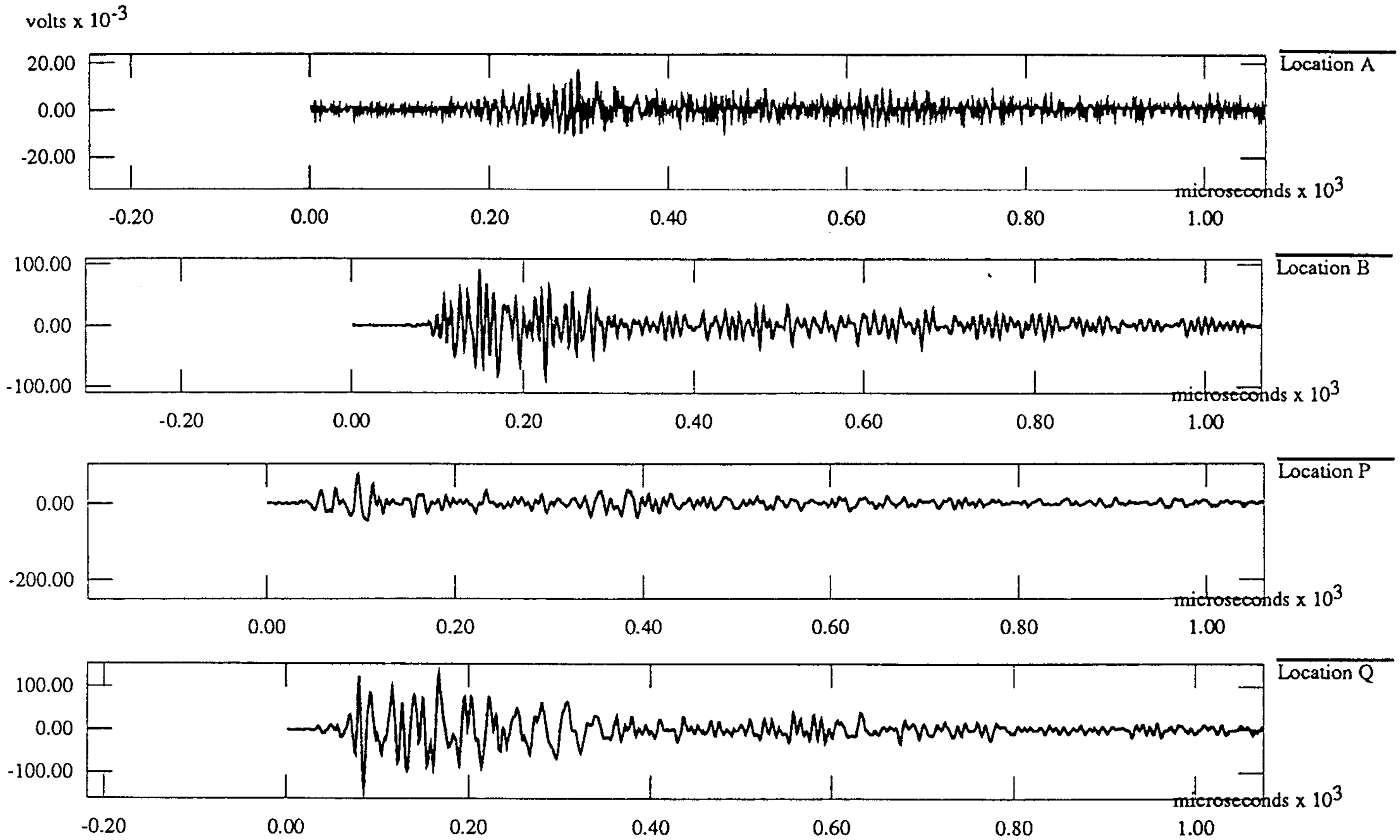

Figure 5 - Crack AE waveform recorded at the sensor locations A, B, P and Q.

5. EXPERIMENTAL RESULTS

Tables 1-3 summarize the results of the classification experiments described above. The number of classification errors on the testing sets were recorded for each test in the experiment. The recognition rate of the network classifier was computed by taking the average of the crack and fretting AE returns correctly classified. As can be seen in Table 1, the neural network classifier yielded perfect recognition rates for the classification of crack growth and fretting AE events from a single sensor location. The results of the sensor location dependent experiment, summarized in Table 2, suggest the network classifier was able to generalize from the AE returns from various sensor locations and was able to classify AE returns in the test set with a high degree of accuracy. Visual inspection of the AE returns used in the sensor location dependent test number four revealed that three of the five returns that were misclassified were exceptions in that the signal had very low amplitude and showed the effects of AE digitization which took place in data collection. In the other tests in this experiment, at least one of these patterns was present in the training set and the network was able to learn features useful for correctly classifying them.

Test	Number of errors	Recognition Rate
1	0	100%
2	0	100%
3	0	100%
4	0	100%
5	0	100%
6	0	100%
7	0	100%

Table 1 - Single sensor series

Test	Number of errors	Recognition Rate
1	1	95%
2	2	91%
3	2	91%
4	5	78%
5	0	100%
6	1	95%
7	0	100%

Table 2 - Sensor location dependent series

Test	Number of errors	Recognition Rate
1	6	74%
2	2	91%
3	6	74%
4	7	69%
5	2	91%
6	3	87%
7	3	87%

Table 3 - Sensor location independent series

Tables 1 - 3. Summary of the network classification experiments.

Comparison of the results shown in Table 2 to those in Table 3 indicates the performance of a network trained on examples from all sensors was consistently better than a network trained on AE returns from arbitrarily selected sensors. The lower recognition rate in the independent series can be attributed to particular training/test set pairs, where AE returns from a sensor location appeared in the test set but were not represented in the training set and were misclassified. This indicates that AE returns associated with specific sensor locations are important for accurate classification.

However, inspection of the training and test sets used in the independent experiment series also suggest the network does not require a large number of examples from a sensor location to correctly classify AE returns. For example, in test number five of the sensor location independent series, the network correctly classified eight AE returns from sensor location P, yet only seven of 150 returns in the training set were from this sensor location. Overall, the results from these experiements were very encouraging.

6. STRATEGIES FOR IMPROVED LEARNING AND GENERALIZATION

As stated in section 2.3, the backpropagation learning procedure is computationally expensive and, as a result, it is difficult to accurately train the neural network classifier within a reasonable period of time. This problem is directly tied to the large number of AE examples required to train the network and the size of the network. There are numerous variations on the backpropagation learning procedure that are designed to accelerate the rate of learning. We experimented with several of these variations and found that while many accelerate the learning rate by as much as an order of magnitude or more[10], the generalization performance of the network actually degraded. Thus, we were careful to select variations in the learning procedure which were robust, fast and maintained the generalization performance of the network. In fact, we were pleasantly surprised to find that the acceleration techniques we chose actually improved the generalization performance of the network.

6.1 Alternative learning strategy

In order to achieve the optimal classification accuracy from the MLP, it is necessary to ensure that there are a sufficient number of example patterns in the training set and that the network learns from each of them. In this way, the network is encouraged to discover regularities in the example patterns, and to de-emphasize those features that turn out to be merely irrelevant idiosyncracies of a few patterns. The learning strategy describes the way in which the training samples are presented to the network.

The typical backpropagation learning strategy is to present all of the example patterns in the training set to the network, and then allow the learning algorithm to adjust the connection weights. After performing numerous simulations of the learning procedure, we made the empirical observation that the efficacy of this learning strategy decreases as the size of the training set increases. This is particularly evident when there is a great deal of variation among the examples of a particular class of AE returns. Intuition suggests that this learning strategy has difficulty in detecting useful features and regularities when presented with numerous examples, since evidently the features learned from each pattern tend to get lost in the shuffle. This observation suggests that a more efficient learning strategy would be to gradually introduce the examples in the training set to the network.

In our learning strategy, two patterns are randomly selected from the training patterns available for each class. These patterns form the initial training set for the MLP network. Backpropagation learning then proceeds to improve the classification performance of the MLP using this limited training set. Given such a small training set, the learning procedure is able to learn the examples quite rapidly. As we might expect, the generalization performance of the network is quite low since it has only been exposed to a small percentage of the training patterns. A new training set is built containing twice the number of patterns, which are selected at random from the available training patterns, and the learning procedure is continued until, finally, the training set contains all the available training patterns.

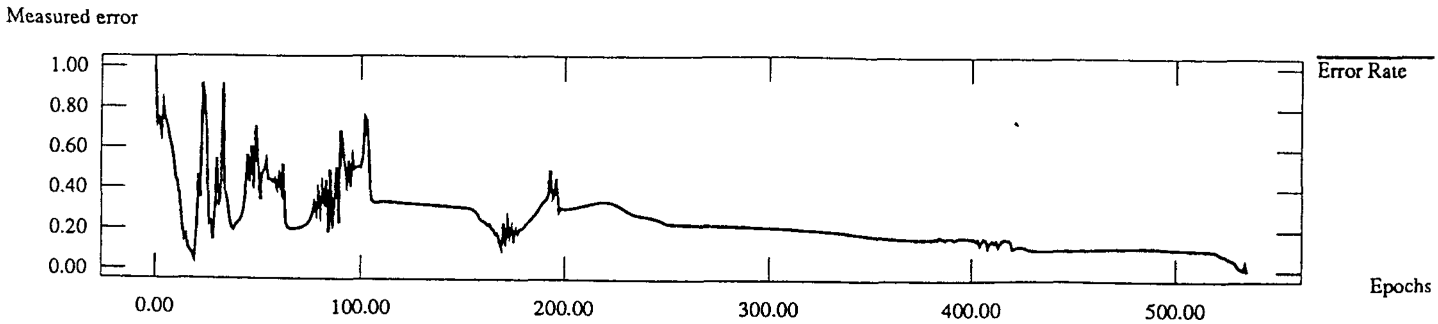

Figure 6 - Measured error versus number of training epochs (increasing the training set size).

Fig. 6 shows the progress of the learning strategy during the training of the MLP network for the crack-fretting experiment. The measured error is 1/2 the squared error of the two output processing elements, normalized for the number of training samples. In this run, the number of samples used were 4, 8, 16, 32, 64 and 112. As can be seen from Figure 6, the error briefly jumps up every time more variability is introduced by way of more training patterns. The network is then forced to improve its performance and to discover features that generalize better and de-emphasize those features that are merely irrelevant idiosyncracies of the limited training set. This learning strategy allows the network to rapidly discover a set of useful features and regularities in the training patterns, before the rather slow fine tuning of the features over all training samples begins.

6.2 Alternative network architectures

After performing a few early simulations of our network architecture, we observed after training the receptive field of processing elements (recall from section 2.1, that the *receptive field* of a processing element is the set of processing elements from which it receives input signals) in the first hidden layer were relatively small regions of the input layer of the network. In effect, the processing elements in this first hidden layer were using only a small portion of their available connections. Moreover, a processing element in the first hidden layer was only "monitoring" a relatively small region of the AE waveform, though collectively the receptive fields of all units in the first hidden layer covered the entire waveform.

Given this insight of how the network was processing the AE waveform, it was natural to constrain the receptive field of the processing elements in the first hidden layer to a small subset of processing elements in the input layer of the network. We experimented with various structures of receptive fields and found *local receptive fields* to be the best. In a local receptive field, the input to the processing element is constrained to a small *local* subset of the processing elements in the previous layer and, as such, local receptive fields preserve the topographical mapping between the AE waveform and the first hidden layer of processing elements.

7. DISCUSSION

In this paper we have presented a neural network approach to the classification of acoustic emission (AE) signals in a simulated airframe joint specimen. Three series of experiments were conducted to evaluate the performance of the neural network classifier: AE signal discrimination, sensor location independent discrimination and sensor location dependent discrimination. The AE signal discrimination series were designed using training and test sets from a single sensor location, with the objective of evaluating the feasibility of our approach. The sensor location dependent series were designed to ensure that returns from all sensor locations in the total set of returns were represented in both the training and test sets with equal frequency. Our object was to evaluate the ability of the neural network to select robust features to enable the classification of AE signals from multiple sensor locations. The sensor location independent series were designed with training sets selected at random from available AE returns, and test sets were designed to ensure that returns from all sensor locations would be evaluated.

Through these three experiments with the neural network classifier we have demonstrated three desirable properties related to the classification of AE returns. First, it can automatically discover features and patterns of interrelated features which serve to define the corresponding class of an AE signal. Examples in the crack-fretting classification series demonstrated the basic ability of the network to discriminate between the two AE signals for a single sensor location. Second, the classifier is shift invariant, that is, the features learned by the network are insensitive to small shifts in time. The sensor location independent experiments demonstrate that the network classifier was indeed able to learn AE features from varying signals and use them effectively to classify signals from different sensor locations. Third, it can generalize to novel AE returns. In the sensor location dependent experiments we evaluated the network performance on waveforms from different sensor locations.

In addition, we have demonstrated how local network architecture and learning strategies can significantly improve learning, by both reducing the number of training epochs required and improving the generalization performance of the network. However, the development of the network architecture and learning strategies were driven not by a theoretical analysis, but by observing problems that occurred in backpropagation learning and by attempting to cure these problems one by one. Now that we have seen what can be accomplished, it would be useful to try to develop a theoretical understanding of some of these tricks. For example, it might be possible to develop a better understanding of how best to present patterns from the training set to a neural network. This sort of understanding should ultimately lead us to more elegant theories of learning in artificial neural networks.

Although this is a limited study in many respects, the results suggest that neural network classifiers should provide a viable alternative to existing techniques for classifying AE returns. We have tried several variations of network architectures and training strategies, but many more variations, including alternative representations of the AE return, are conceivable. Some of these variations could potentially lead to significant improvements over the results presented in this study. Our goal here is to present neural networks as a new and promising approach for AE signal classification. Their power lies in their ability to develop shift invariant features of acoustic waveforms and use them in making optimal decisions. This holds significant promise for AE classification in general, as it could help overcome the representational weaknesses of recognition systems faced with the uncertainty and variability in real-world signals.

8. ACKNOWLEDGMENTS

The authors would like to express their gratitude to the Computational Sciences Department and the Automation and Measurement Sciences Department of Pacific Northwest Laboratory (PNL) for their support of this project. We would also like to thank the members of the Knowledge Based Systems Group at PNL, especially Debbie Gracio and Mark Whiting, for their constant help in the various stages of this research and in the preparation of this manuscript.

9. REFERENCES

1. M.A. Friesel, "Application of Signal Analysis to Acoustic Emission from a Cyclically Loaded Aluminum Joint Specimen," *Journal of Materials Evaluation*, Vol. 47, pp. 842-848, 1989.

2. R.B. Melton, "Classification of NDE Waveforms With Autoregressive Models," *Journal of Acoustic Emission*, Vol. 1, pp. 266 - 270, 1982.

3. R.P. Lippmann, "An Introduction to Computing With Neural Networks," *IEEE ASSP Mag.*, Vol. 4, pp. 4-22, April 1987.

4. I.D. Longstaff and J.F. Cross, "A Pattern Recognition Approach to Understanding the Multi-layered Perceptron, *Pattern Recognition Letters*, Vol. 5, pp. 315 - 319, 1987.

5. D.E. Rumelhart and J.L. McClelland, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. I and II. M.I.T. Press, Cambridge, MA., 1986.

6. R.S. Barga, "Artificial Neurocomputing for Data Primitive Extraction," *Proceedings of the 4th Annual Aerospace Applications of Artificial Intelligence Conference,* Vol. 1, pp. 106-117, Dayton, Ohio, 1988.

7. R.S. Barga and R.B. Melton, "Framework for Distributed Artificial Neural Network Simulation," International Joint Conference on Neural Networks, Vol. 2, pp. 94-97, Lawrence Erlbaum Associates, Inc., Washington DC, 1990.

8. G. Murray, "A Cautionary Note on Selection of Variables in Discriminant Analysis", *Applied Statistics*, Vol. 26, pp. 246 - 250, 1977.

9. G.T. Toussaint, "Bibliography on Estimation of Misclassification," *IEEE Transaction on Information Theory*, Vol. 20, pp. 472 - 479, 1974.

10. R.A. Jacobs, "Increased Rate of Convergence Through Learning Rate Adaption," *Neural Networks*, Vol. 1, pp. 295-307, 1988.