Chapter 7

Structural Health Monitoring: Inverse Problem

7.1 Introduction

In the previous chapter (Chapter-6) it has been seen that if the location and the size of damage is known, the finite element analysis can calculate the structural responses or sensing voltage in sensor from known tip load or actuation current. But in Structural Health Monitoring (SHM) it is more important to get the damage status (if exist) from actuation current and sensing voltages. Broader sense it is an inverse problem. Most of the inverse problems generally require the solution of an ill-posed system of equations. Especially for a problem in which the number of unknown parameters exceeds the measured data, it is difficult to identify the unknown parameters. The identification of damage properties from sensor responses is very often ill-posed due to non-uniqueness of the solution. And this facility is not available in the finite element framework. Hence, researchers are looking for alternate approaches for many years. One of the simplest approaches is by trial and error. Steps for trial and error approach consists of - (1) assume one damage configuration in the structure, (2) analyze the structure with this damage configuration using finite element analysis and get the responses of the structure, (3) check these responses with the desired responses and if these responses doesn't match with the desired values return back to the step (1). The main drawback of this approach is that it requires a number of finite element analysis to find the exact damage configuration due to the absence of suitable procedure to use the experience of previous trials. Moreover, the computation cost involved is exorbitant. This trial and error approach is quite laborious unless a good initial guess is available. The Artificial Neural Network (ANN) is one such method, which gives a mapping relationship between the responses of the structure and the damage configuration of the structure, which can be used to get good assumption for trial and error approach in the SHM.

In recent years, some researchers have applied ANN for structural damage detection related applications [175, 215, 257, 294, 348, 350, 351, 358, 401, 402, 407, 416, 419, 418]. Much experience has been accumulated through these exploratory investigations. Lee et al. [215] used the differences or the ratios of the mode shape components between before and after damage for input to the neural networks. Faravelli and Pisano [117] shown neural network approach to identify the damaged element of a multi-bay planar truss structures. Angeles et al. [16] investigated nonlinear system identification of civil structures using ANN. Xu [408] used ANN for sub-structural parametric identification of post-earthquake damaged civil structures by the direct use of dynamic responses. Lee and Lam [214] presented a neuro-fuzzy model for the diagnosis of structural damage. Rhim and Lee [328] examined the feasibility of using ANN in conjunction with system identification techniques to detect the existence and to identify the characteristics of damage in composite structures. Martin et al. [255] used artificial neural receptor system to reduce the number of channel of data acquisition system for SHM. Zang et al. [421] presented structural damage detection from time domain data using independent component analysis and ANN. Su and Ye [349] presented an online health monitoring technique for in-service composite structures using 3D FEM, ANN and wavelet transform technique. Ye et al. [416] reviewed performance of artificial intelligence techniques in functionalized composite structures and as a specific case study, SHM technique was developed through an ANN. Kao and Hung [175] detected structural damage using ANN from free vibration responses. Su and Ye [348] trained multi-layer feed-forward, error-back propagation ANN with the raw Lamb wave signals for SHM and validated the proposed method by locating actual delamination and through-thickness holes in quasi-isotropic composite laminates. Nyongesa et al. [294] proposed a technique combining conventional image analysis, ANN classification and fuzzy logic inference to characterize shearograms for impact damage detection.

In this chapter a methodology for automatic damage identification and localization in composite structural components is presented using ANN. Responses for delaminated and healthy beam are computed using finite element analysis, which is discussed in the earlier chapters. These responses and corresponding damage configurations are used to train ANN as input and output of the network respectively. A feed-forward back-propagation neural network is designed, trained, and to predict the delamination location using the structural response as inputs. Time domain response history is represented by the responses of every time steps, which is high dimensional. Hence, dimension reduction technique is essential to use the response in the network. In Section-7.2, single ANN is used to predict the damage properties after dimension reduction of time domain responses. This dimension reduction looses the output uniqueness of the mapping from response histories to damage properties. To overcome output uniqueness problem, output space is partitioned and an ANN is trained for each partition. Finally, these trained ANNs are coordinated with a committee machine, which is illustrated systematically in Section-7.3. In Section-7.4, a hierarchical neural neural (HNN) is trained, validated and used for mapping of the damage properties from response histories of the structure. Hundreds of ANN is coordinated systematically with this HNN.

7.2 SHM using Single ANN

An artificial neural network (ANN) can learn the mapping between inputs and outputs through a sample data set and determine the output of new data based on previous knowledge. In this study, ability of identifying structural damage status for an ANN is acquired through training the neural network using the known samples. Data obtained from the magnetostrictive sensors described in earlier chapters, is used to train the network to identify the delamination size of the composite laminate. The multi-layer feed-forward network that is the most popular of the architectures currently available, is used. This is a supervised learning procedure that attempts to minimize the error between the desired and the predicted outputs. Standard logistic function $y = 1/(1 + e^{-1.7159v})$ is applied in the hidden layer as activation function with linear output layer. This algorithm adjusts the connection weights according to the back-propagated error computed between the observed and the estimated results. The sample data set is separated into two parts, one for training and the other for testing the network performance. The training stage involves the network weights being adjusted until the network is able to satisfactory map all data. Once trained, the network provides rapid mapping of a given input into the desired output quantities. This, in turn, can be used to enhance the efficiency of the mapping process.

Figure-7.1 shows the delaminated, 12 layered unidirectional composite cantilever beam with actuator and sensor. Sensor response due to actuation is computed for the beam as per the analysis shown in earlier chapters. In this section, numerical experiments

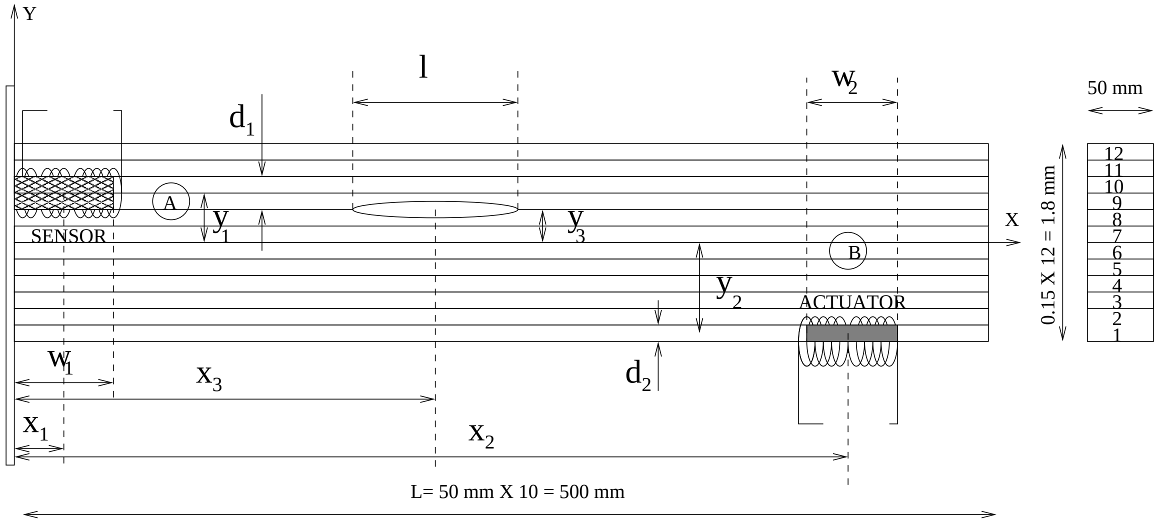


Figure 7.1: Delaminated composite beam with sensor and actuator.

are performed to identify the size of delamination using single ANN, where delamination layer is known. As structural damage information is distributed in different vibration modes, and vibration modes with high frequencies are generally more sensitive to small damage, frequency of 5kHz is considered to excite the actuator (shown in Figure-3.6(b)) for better performance in the identification procedure. 51 different sample data sets are numerically simulated for each layer of the delamination. These 51 cases include the intact laminate and laminates with delamination of 50 different sizes (10 mm to 500 mm with 10mm increment), where delaminations are started from support of the cantilever. In order to identify the delamination length at each layer, one BP neural network with 10 inputs and 1 outputs are designed. Output node denotes the length of the delamination. Every layer networks are trained by their corresponding 51 sample data. These samples are for delamination in the corresponding layer or for healthy structure. Thus there are eleven neural networks and these are trained for the corresponding layer to predict the size of the delamination. Open circuit voltages of the sensor are high dimensional data. So, first 5 peak values and their locations in time history are taken as input space of the ANN mapping. Out of 11, two experiments are shown- one for top layer delamination and other for mid layer. For each experiment two networks are trained with 10-5-1 (Figure-7.2) and 10-5-2-1 (Figure-7.3) architectures. All these four networks are trained up to one million epochs and Figure-7.4 shows the training performance of the networks. With architecture of 10-5-2-1, both the networks for top layer and mid layer are trained successfully. But training performance of top layer networks with 10-5-1 architecture is relatively poor.

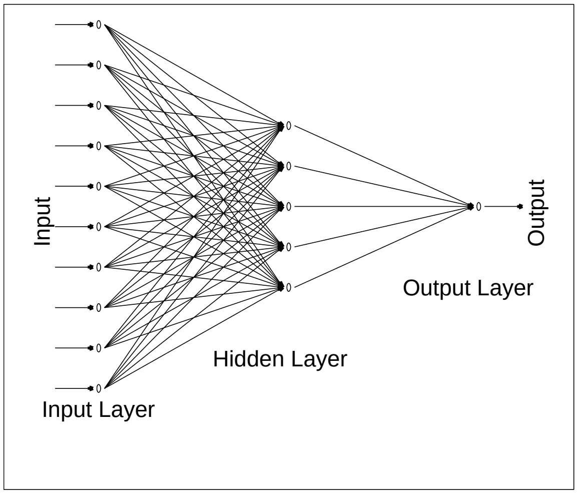


Figure 7.2: ANN of 10-5-1 Architecture.

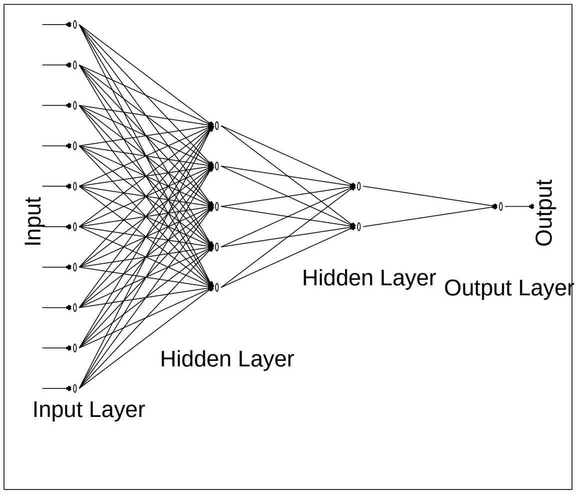
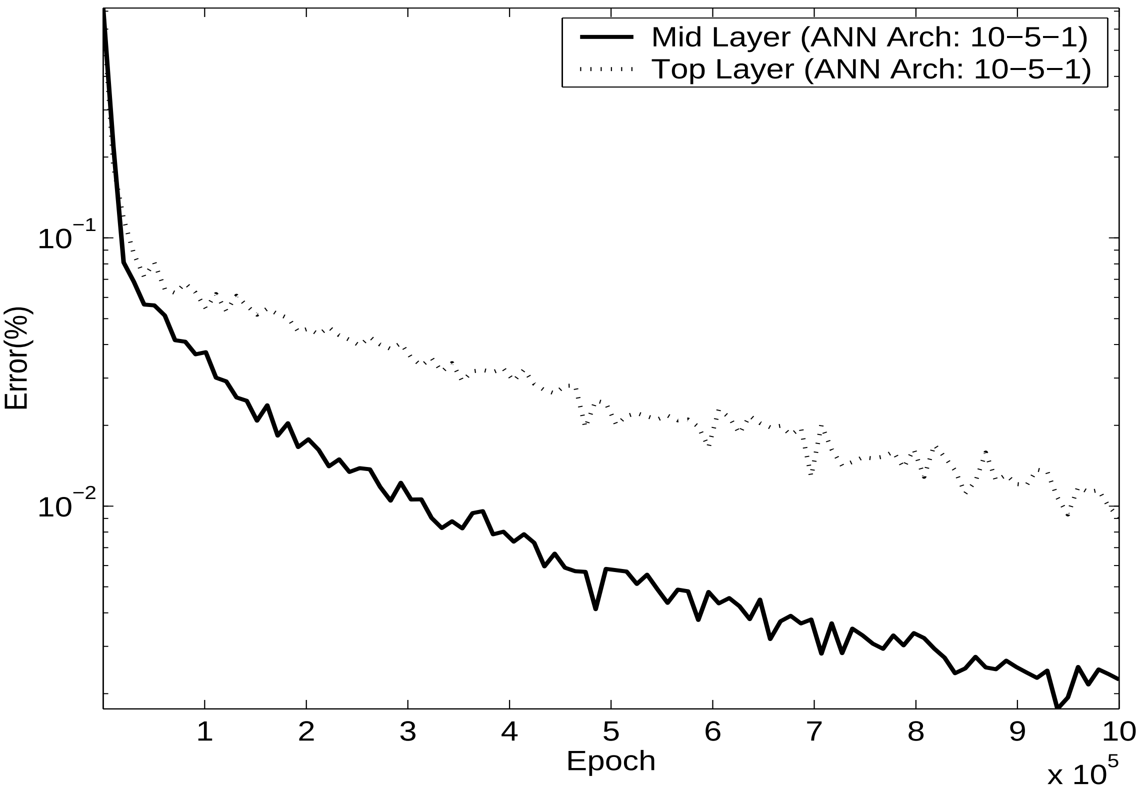


Figure 7.3: ANN Architecture 10-5-2-1.

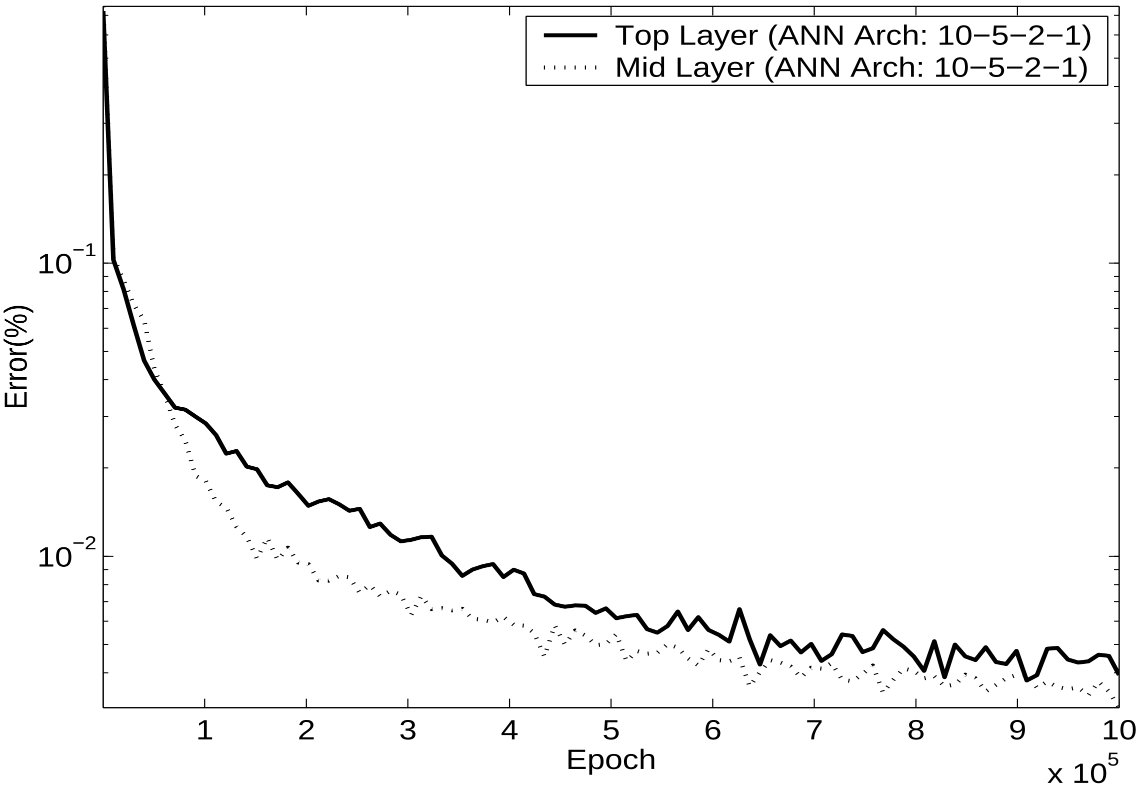
After training, networks are tested with different set of example for identification of generalization capacity. 50 different sample data sets are numerically simulated for each layer of the delamination for testing of these trained networks. These are for delamination of size 5 mm to 495 mm in 10mm increment. Corresponding testing performance of trained networks are shown in Figure-7.5, where testing error gives an indication of the generalization capacity of the trained network. For 10-5-1 architecture, both top and mid layer networks are trained with satisfactory generalization capacity. Where as generalization capacity of the mid layer network with 10-5-2-1 architecture is poor, which may be due to overtraining of the network. However, it is showing that for both the architecture, delamination is identified within 50mm accuracy by any of the trained networks.

7.2.1 Difficulties in Single ANN

Main problem of these trained 11 ANNs are, the delamination layer should be known before, such that corresponding trained ANN will be used to predict the size of delamination. This problem can be overcome by using single ANN with two output nodes one for size and other for layer of the delamination. Combining the layer and the size of the delamination, one single network is trained with two output node as per Figure-7.6. For this experiment, one single ANN is tried to train with all these samples. However, this network fails to train the samples, which is due to similar responses for different delamina-

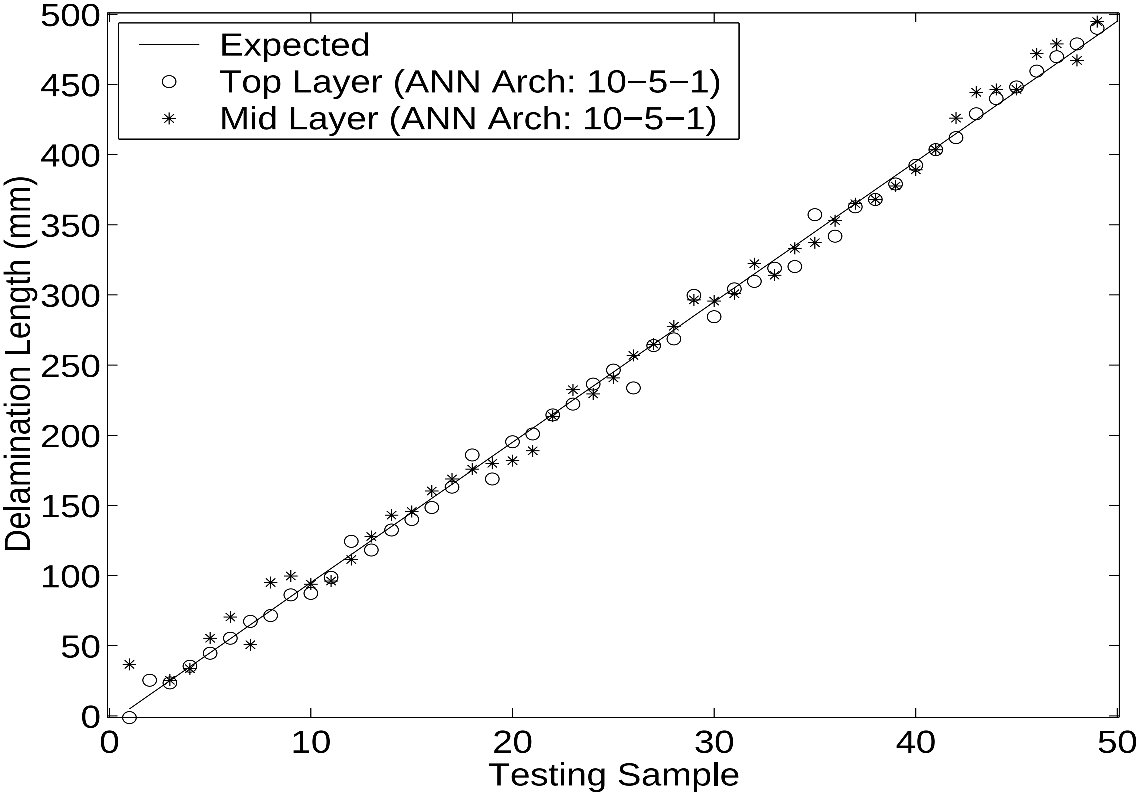


(a) 10-5-1 network architecture

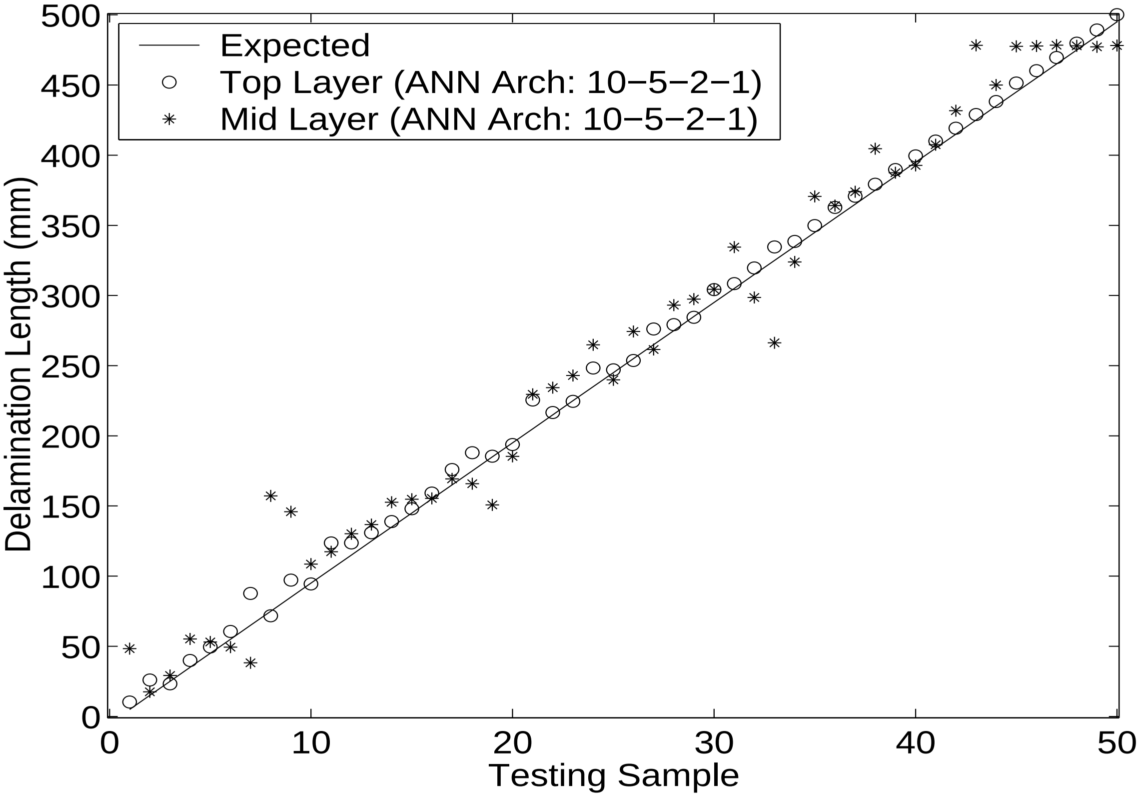


(b) 10-5-2-1 network architecture

Figure 7.4: Training Performance of ANNs.



(a) 10-5-1 network architecture



(b) 10-5-2-1 network architecture

Figure 7.5: Testing Performance of ANNs.

tion configuration in dimensionally reduced input space. For training of networks, output space uniqueness is essential requirement. Hence, to keep output uniqueness in the ANN mapping, the output space is divided with different overlapped zones. Samples from these zones are used to train a corresponding ANN with two output nodes. These ANNs are coordinated with one committee machine, which is illustrated systematically in the next section.

7.3 Committee Machine

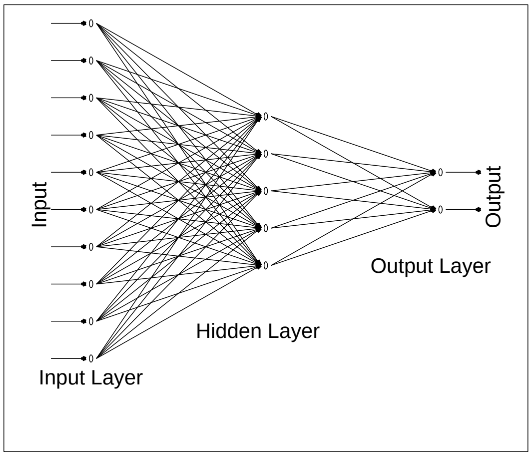


Figure 7.6: ANN Architecture 10-5-2.

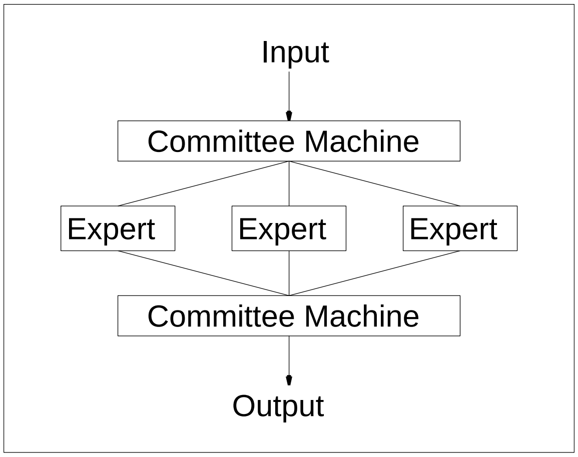


Figure 7.7: Committee Machine.

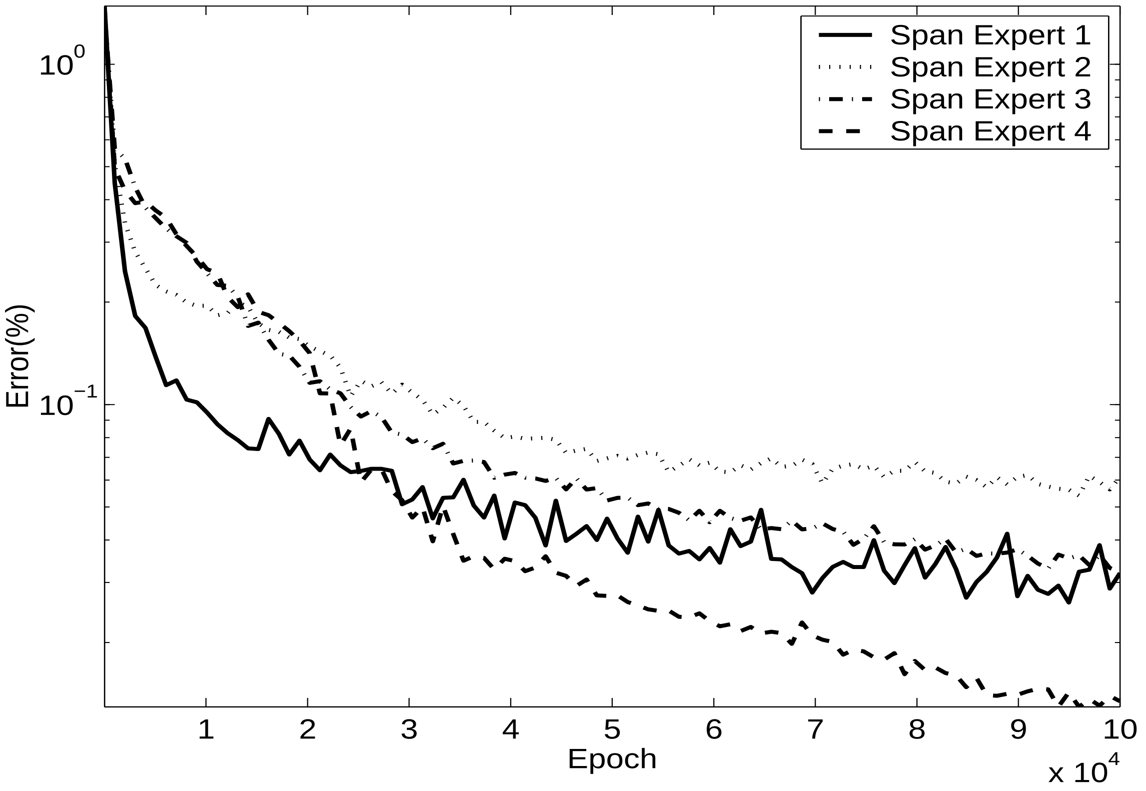
It is perhaps impossible to combine simplicity and accuracy in a single model of ANN. Here a simple model that uses hard decisions to partition the input space and output space into a piecewise set of subspaces, with each subspace having its own expert (Figure-7.7). Sizes of these subsets are such that the output uniqueness within a subset is preserved as well as sufficient number of sample data is available to train and test the network. Thus, for every subset, one expert is trained and tested taking sample data from these subsets. Similar to the division of output space, input space can be divided in different subsets on the basis of sensor, actuator and actuation combination. However, as the input space subdivision is known a'priori, committee machine for input space subdivision is not required. Single multi-layered perceptron (MLP) uses a black box approach to globally fit a single function into the data, thereby losing insight into the problem. Here computational simplicity is achieved by distributing the learning task among a number of experts, which in turn divides the input space and output space into a set of subspaces. The combination of experts is said to constitute a committee machine. Basically, it fuses knowledge acquired by experts to arrive at an overall decision that is supposedly superior to that attainable by any one of them acting alone.

7.3.1 Numerical Experiment using Committee Machine

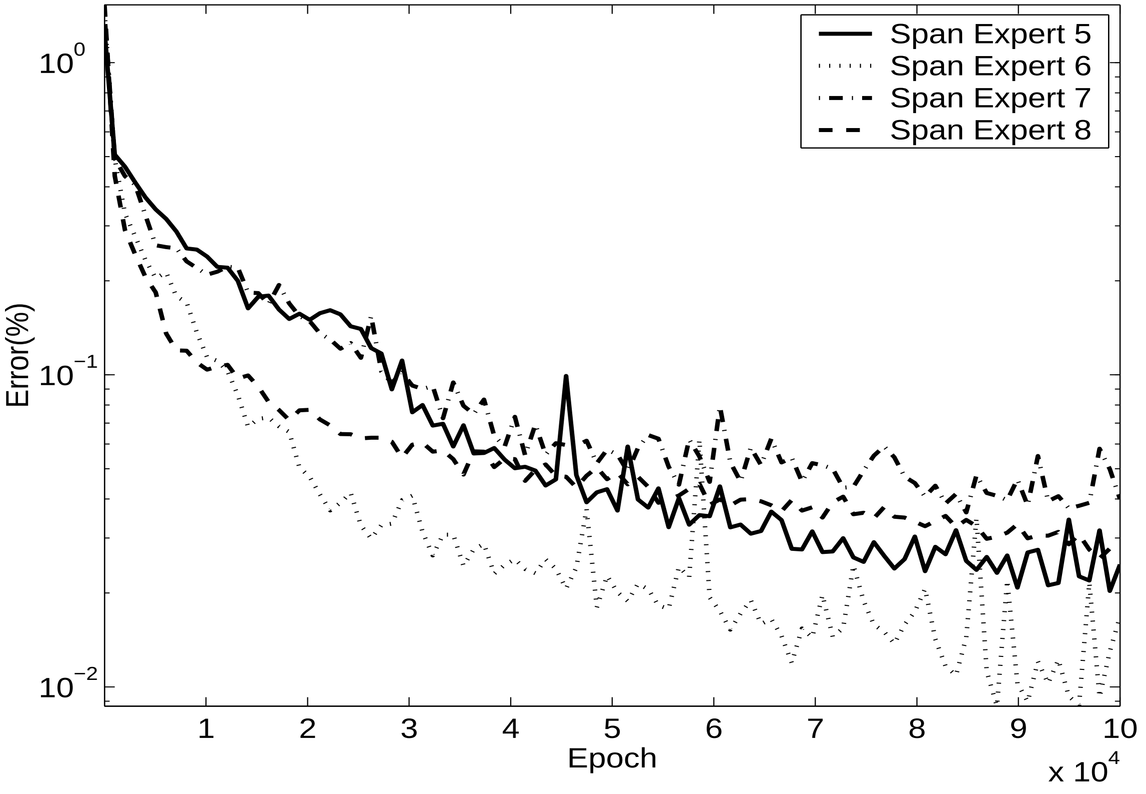
Table 7.1: Performance of Committee Machine.

Experts	Span Left S_l(mm)	Span Right S_r (mm)	Layer Opinion O_l(mm)	Span Opinion O_s(mm)	Distance D
1	0	50	-2.21	11.5	1.25
2	25	75	1.82	62.1	0.77
3	50	100	1.02	174	3.67
4	75	125	-2.13	92	0
5	100	150	-1.95	173	0.93
6	125	175	-1.61	168	0.55
7	150	200	8.64	172	22.77
8	175	225	-0.721	181	0
9	200	250	-2.63	238	1.88
10	225	275	-4.32	230	5.51
11	250	300	-0.444	314	0.35
12	275	325	-0.554	308	0
13	300	350	-1.01	322	0.07
14	325	375	0.203	334	0
15	350	400	-0.592	380	0
16	375	425	-0.595	380	0
17	400	450	0.600	458	0.19
18	425	475	0.0818	403	0.62
19	450	500	0.730	494	0

The beam is divided span wise in 19 different overlapped zones. For each zone, one network (expert) is trained to predict the delamination layer as well as location. Table-7.1 shows the span wise location in second and third columns (S_l, S_r) for every expert. Training and testing samples for corresponding experts are taken from their corresponding zones. Figure-7.8 shows the training performance of different span wise experts. Networks are trained up to 10^5 epoches. After that, the training errors become saturated. Figure-7.9 shows the training performance of the first and fifth span wise experts. Both predicted

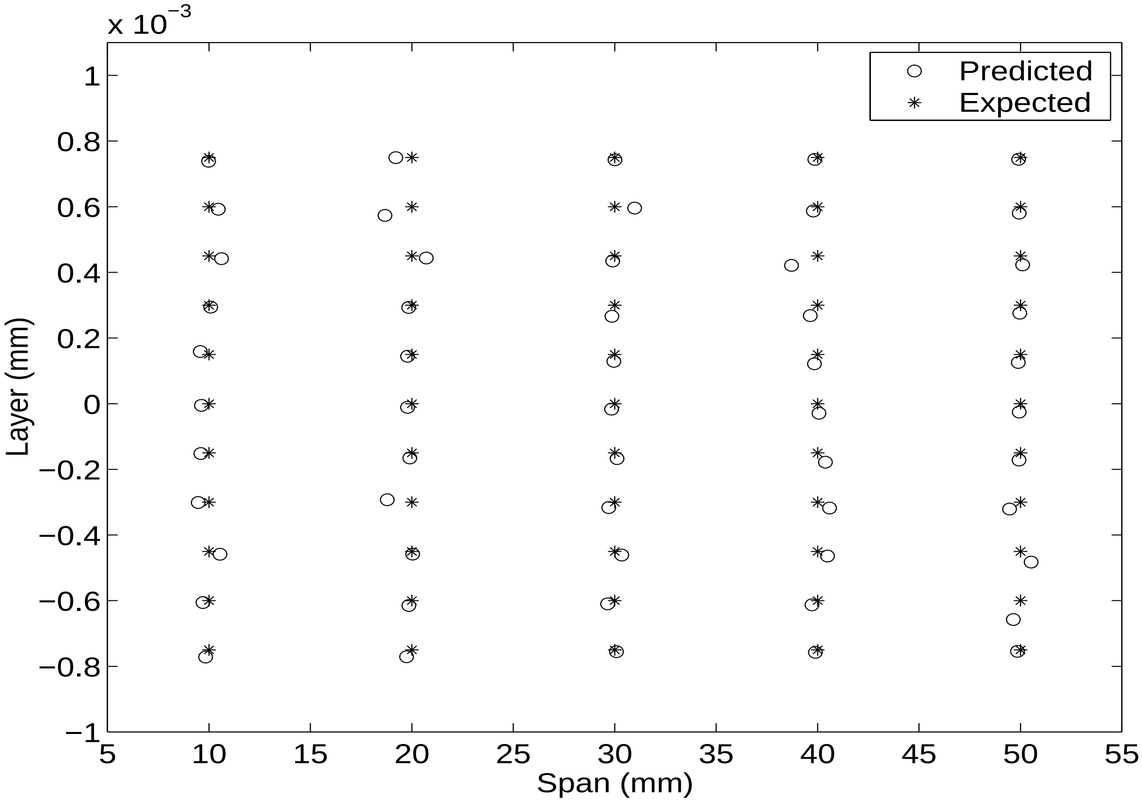


(a) Span Expert 1-4

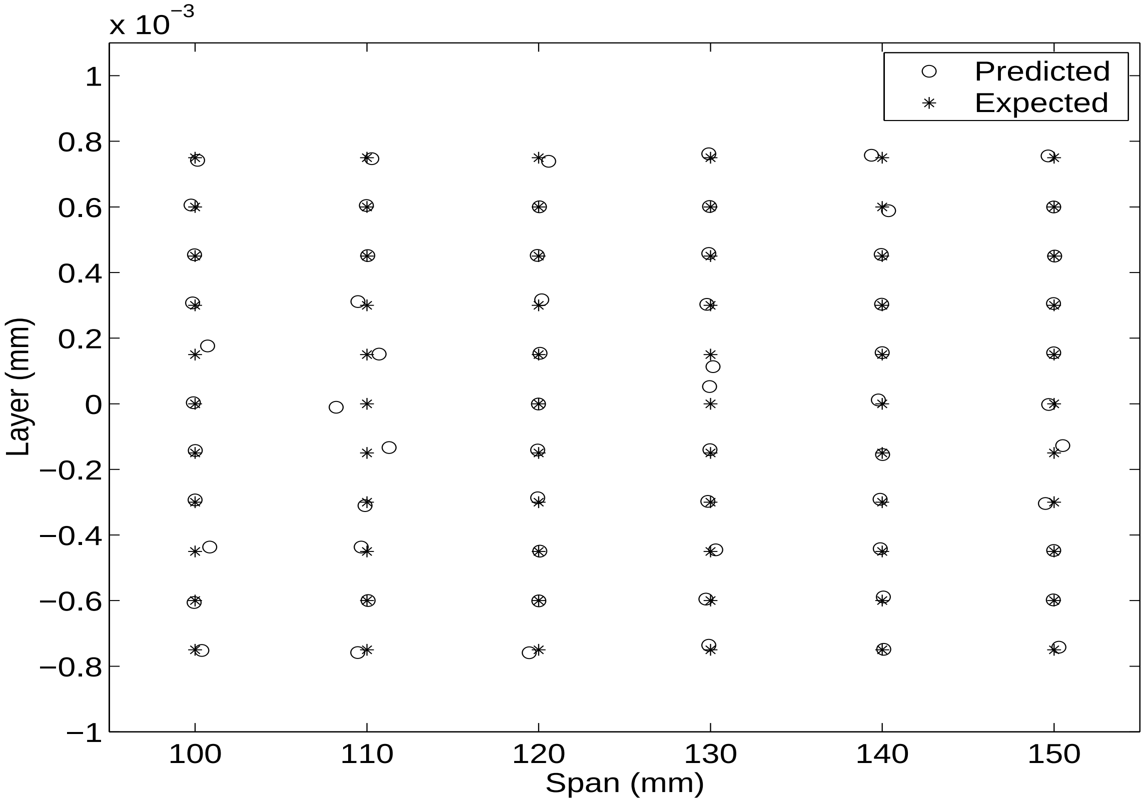


(b) Span Expert 5-8

Figure 7.8: Training Performance of Different Span Experts.



(a) span expert 1



(b) span expert 5

Figure 7.9: Training Performance of span experts

and expected layer and size of the delamination is shown in the figure. Layer wise errors of the training samples are with in the thickness of the composite layer and span wise errors are with in 5mm. Hence, this can be considered as well trained. After training, each of the networks is used to test the generalization capacity as discussed in the earlier section.

In identification face, time domain sensor outputs are fed into every networks to get their opinions like, the location and size of the delaminations. These individual opinions are fused by committee machine, which is illustrated in Table-7.1 through a simple example. Table shows the performance of the committee machine for a delamination configuration of (-0.6,380). In delamination configuration, first coordinate is for layer and second is for delamination size. Prediction of the Expert-1 (which is trained with samples from span of 0mm to 50mm) for this delamination configuration is (-2.21,115). Similarly, every expert has given their opinions, which are shown in fourth and fifth columns in the Table. As the networks are trained with the samples taken from their corresponding zones, networks are experts for their corresponding zone only. Hence, the location of these predicted opinions are checked with their corresponding experts zone first. For that, a distance norm is used, where distance between their opinion and expertise zone is defined as:

$$D = Max\left[\frac{(O_s - S_l)(O_s - S_r)}{(S_r - S_l)(S_r - S_l)}, \quad \frac{(O_l - L_b)(O_l - L_t)}{(L_t - L_b)(L_t - L_b)}, \quad 0\right]. \tag{7.1}$$

O_s and O_l are the opinion span and layer respectively. In the expression, top of the layer (L_t) and bottom of the layer (L_b) are considered as +0.9mm and -0.9mm respectively. If the distance is zero, the opinion is with in its expert location, and can be used as an active expert for this particular delamination configuration.

In the Table-7.1, only expert-4, 8, 12, 14, 15, 16 and 19 are the active experts as their opinions are with in their corresponding expert zones. However, only expert 15 and 16 gives mutually agreeable prediction and all other give mutually contradictory opinion. Therefore, these noisy experts are removed from the active list and average opinion of the expert 15 and 16 is used as prediction of the committee machine. Average of these opinion is (-0.593,380), which is very close to original delamination configuration (-0.6,380).

When the number of active expert is quite large, finding of mutually agreeable experts or removing of noisy expert is difficult manually. One systematic procedure to remove such noisy experts is explained in Figure-7.10. All the active experts are plotted according to their opinion in the figure. First, the mean of all these active expert's opinion is computed and shown in point 'A' in the figure. Then furthest expert from this mean

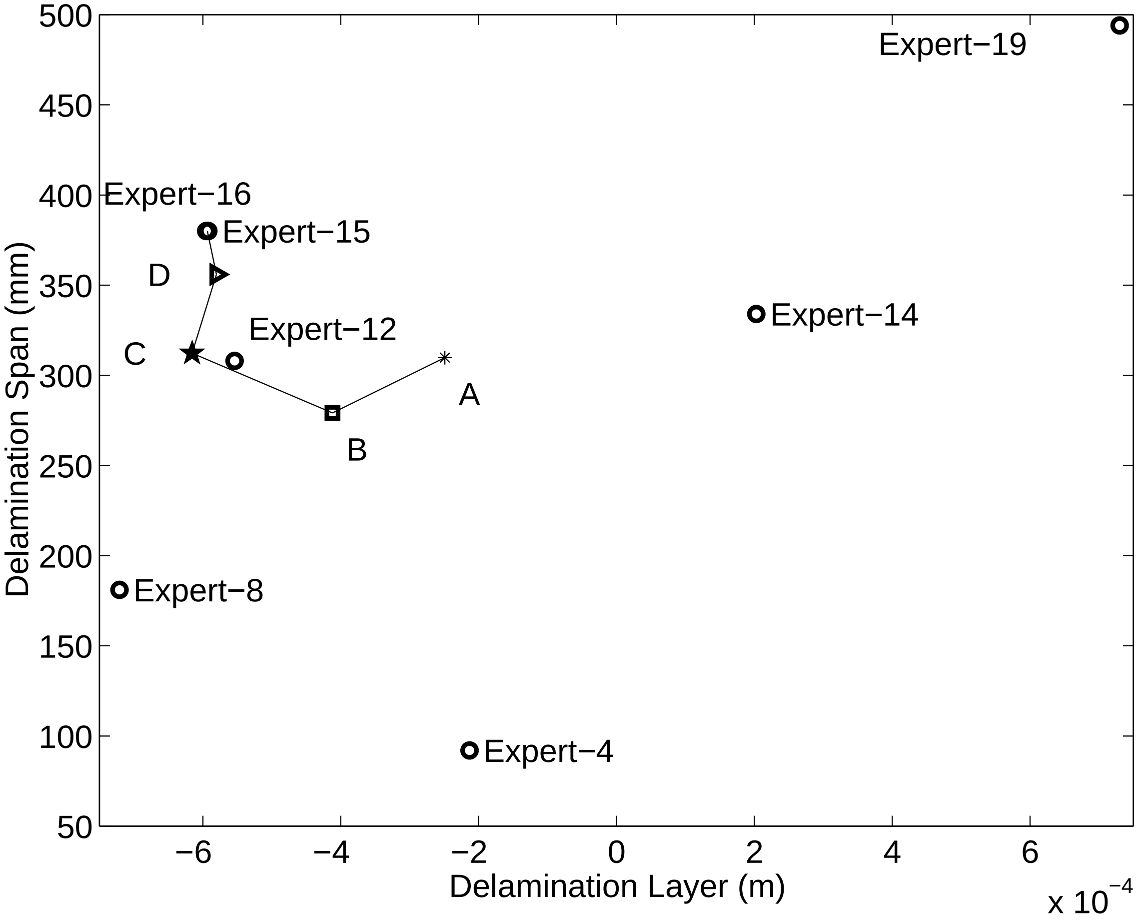


Figure 7.10: Removal of Noisy Expert.

(here Expert-19) is considered as noisy expert and excluded from the list of active expert. After excluding this noisy expert, again mean is calculated among the rest of the active experts, which is shown as point 'B' in the Figure. In this way, gradually all the noisy experts will be removed one by one and mean of the remaining active experts will move from points 'A'-'B'-'C'-'D' to the average opinion of expert-15 and 16.

7.3.2 Information Loss due to Dimension Reduction

Dimension of the time domain response is so large, it is not possible to use directly as input node in the neural network. Moreover, values for some of the nodes are constant for the entire training sample, which will create difficulty to train the network. As an example, responses at time step zero are zero for every delamination configuration. One approach of this problem is dimensional reduction of the responses, like, principle component analysis, independent component analysis, peak values or locations. However, the information loss due to this dimension reduction may reduce the damage detection capacity of the ANNs. This information loss can be compensated using more number of dimension reduction procedure, rather than a single one, where probability of preservation of the damage signature is more. Next section gives a systematic procedure to use more than one dimension reduction in hierarchical Neural Network (HNN).

7.4 Hierarchical Neural Network (HNN)

With the common multi-layer neural network architectures, networks lack internal structure; as a consequence, it is very difficult to discern characteristics of the knowledge acquired by a network in order to evaluate its reliability and applicability. Alternatively, neural network architecture is presented, based on a hierarchical organization. By using the hierarchical scheme, a complicated large-scale system is decomposed into a set of lower order subsystems and a coordination process, and thus becomes tractable.

7.4.1 Five Stage HNN

A five-stage hierarchical neural network is designed by combining multi-layer perceptrons for first stage and mixture-of-experts in the subsequent stages. The second stage mixture-of-expert, ensembler network, learns to minimize the overtraining errors. The third stage mixture-of-expert, validation network, learns to minimize the generalization errors. The fourth stage mixture-of-expert, expert network, learns to minimize the error of network due to loss of information for reduction of input space dimension, and finally, the fifth stage committee machine choose the appropriate expert network from all expert networks. The whole procedure is discussed as follows.

7.4.1.1 Neural Networks

In the first stage, multi-layer perceptrons are used to map dimensionally reduced sensor response to damage space. Training and testing of a single multi-layer perceptron is explained in Section-7.2 extensively.

7.4.1.2 Ensemble Network

Often artificial neural networks are prone to overtraining, where network trains the computational and experimental noises. And there is no direct rule to draw the line between well trained and overtrained for a set of training examples and network architecture. One of the indirect ways to get a measure of overtraining of the network is Ensemble Network. In ensemble network, a number of neural networks are trained with the same training samples but with different initial conditions, learning rate, training algorithm, network architecture and training sequence (for sequential learning). In training phase, each network trains and generates training error for the training samples. On the basis of these training errors, weightages of the trained neural network is determined, where less training error gives more weightage of the neural network. In the execution phase, these weightages are used to get weighted average of all neural networks output as the output of ensemble network. From the distribution of the output of different neural networks and their corresponding weightages, one measure of overtraining can be computed. If the outputs of different neural network are close (at least for those have more weightages) then networks are well trained otherwise it is overtrained. Every trained neural network is tested through the test data set, which gives the testing errors as a measure of generalization of the neural networks. These testing errors with the weightages of the neural networks give the weighted average of testing error, as a measure of testing error of the ensemble network. Next issue in the HNN is the generalization of the network using the testing error of ensemble network.

7.4.1.3 Validation Network

As, every neural network within the ensemble network is trained with same training sample; these neural networks need testing for generalization of the ensemble network. For testing of network, sample data set is separated into two groups, one for training and other for testing (validation). Different partitioning of sample data is explained in Figure-7.11. After training of neural networks, with the training samples, the neural networks simulate the testing sample and get the testing error. These testing errors with their weightages give the measure of generalization of the ensemble network. However, every neural network in the ensemble network is trained and tested through the same set of training data set and testing data set respectively. To get a more general input-output mapping from a set of sample data set, different division of training sample and testing sample is essential. This can be done through a systematic manner using validation network. Validation Network consists of some number of ensemble networks. These ensemble networks are trained and tested with same sample data set with different partition for training data set and testing data set as shown in Figure-7.11. However, every validation network is trained and

Figure 7.11: Partition of Sample Set.

tested by a fixed set of sample data set, which comes from different dimension reduction procedures.

7.4.1.4 Expert in HNN

In time domain structural health monitoring, time histories of the sensor outputs (open circuit voltages) are the original input space for the input-output mapping. As explained earlier, it is not possible to train the network taking full dimension of the original input space (sensor output). To address this issue, different dimension reduction procedures are available in the literature, which can reduce the dimension of the original input space keeping main features of the high dimensional data. However, for structural health monitoring, suitable dimension reduction procedure is not available, which will reduce the dimension of the input space preserving the signature of damage from the high dimensional sensor output. To overcome this problem, a number of reductions procedures are taken to increase the chance of preserving the damage signature in the reduced input data sets. This issue is taken care by a systematic manner in the fourth stage of the HNN.

Expert consists of a number of validation networks, which were trained and tested through same output of a sample data set but with different lower dimensional input of the data set. These different input data sets come from different dimension reduction procedure of the original input set, which is high dimensional sensor output. Hence, every expert is a mapping from sensor output to the damage properties of the structure and performance of the expert depends on the performances of its validation networks, which is trained through dimensionally reduced input sample set.

7.4.1.5 Committee Machine in HNN

Some times, due to this reduction procedure, the mapping looses the output uniqueness, which is essential for the training of neural network. This problem was studied by partitioning the input and output space into a piecewise set of subsets, with each subset having its own expert. Committee machine is discussed extensively in earlier section (Section-7.3) taking experts as a single neural network, which lacks information about the level of confidence of the decisions.

7.4.2 Training Phase of HNN

In training phase, high dimensional samples (original sensor output) are fed to the expert, which are within their expert zone. Every expert gives these samples to their subordinates (validation networks), after assigned dimension reduction. The validation networks then create different set of partition for training and testing samples and transfer these partitions to the corresponding ensembler network. Similarly, every ensembler network gives these training and testing samples to their subordinate, ANN network, for training and testing of the networks. Thus, ANN gets their samples for training and testing of the network from the original high dimensional sensor open circuit voltages. Training, testing (validation) and execution (simulation) phase of the HNN is illustrated in Figure-7.12 and discussed as follows:

7.4.2.1 Training of ANN

In training phase, every multilayer perceptrons (MLP) are trained with their corresponding training samples. These training samples will give their training error as per the

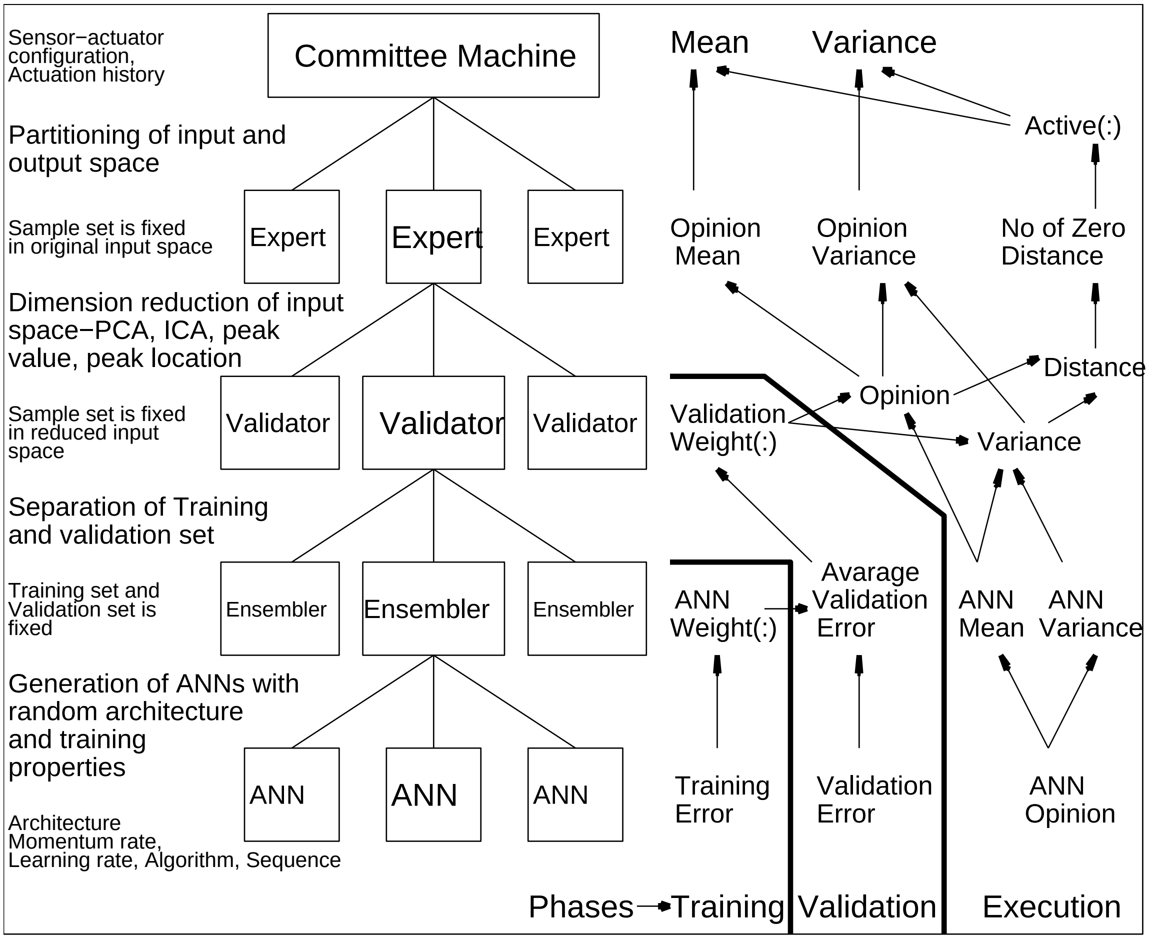


Figure 7.12: Training, Validation and Execution of 5 stage HNN.

discussion in Section-1.4.6.1. Training error for i^{th} ANN, E_t^i, is mean squire error of all the samples used in training. Due to different network architectures, learning algorithms, initial condition, learning rate, momentum rate and learning sequence, the ANNs will be trained differently. This step will take more than 98% computation time of total training of HNN.

7.4.2.2 Training of Ensembler Network

Relative training performances of ANN are stored in Ensembler network. Depending upon the training error, every trained ANN under an ensembler network is associated with their weightages. This ANN weightages (W_A^k) for k^{th} ANN are computed as:

$$W_A^k = \left[E_t^k \sum_{i=1}^{N_a} \frac{1}{E_t^i} \right]^{-1} \qquad \Rightarrow \qquad \sum_{k=1}^{N_a} W_A^k = 1. \tag{7.2}$$

Where N_a is the number of ANN under the ensembler.

This is the training phase of the hierarchical network. Training phase is limited within ANN and ensembler network.

7.4.3 Testing Phase of HNN

As mentioned earlier, in testing phase, every ensembler network gives testing samples to their subordinate ANN network taken from validation network.

7.4.3.1 Testing of ANN

Every trained ANNs are tested with their corresponding testing (validation) samples. These testing samples will give their testing errors as per the discussion in Section-1.4.6.4. Testing error for i^{th} ANN, E_v^i, is mean squire error of all the samples used for testing.

7.4.3.2 Testing of Ensembler Network

Training weightages (W_A) and validation errors (E_v) of all the subordinate ANNs are considered to calculate average validation error (V_E) of an ensemble network. This average validation error for ensembler network is considered as:

$$V_E = \sum_{k=1}^{N_a} E_v^k \; W_A^k \tag{7.3}$$

Where N_a is the number of ANN under the ensembler.

7.4.3.3 Testing of Validation Network

Relative performance of ensembler networks under a validation network are calculated from these average validation errors (V_E). Every validation networks calculate weighted average validation error for ensemble networks. Thus, weighted average validation error (W_V) of k^{th} ensembler network is calculated as:

$$W_V^k = \left[V_E^k \sum_{i=1}^{N_r} \frac{1}{V_E^i} \right]^{-1} \qquad \Rightarrow \qquad \sum_{k=1}^{N_r} W_V^k = 1. \tag{7.4}$$

Where, N_r is the number of ensembler networks under the validation network.

This is the testing phase of the hierarchical network. Testing phase is limited within ANN, ensembler network and validation network.

7.4.4 Execution Phase of HNN

In execution phase, execution samples (high dimensional sensor output) are fed into every expert to get their opinion and confidence. Similar to training phase, execution samples reaches to ANN network after proper dimension reduction.

7.4.4.1 Execution of ANN

ANNs execute these samples and give their opinion (O_a) as discussed in Section-1.4.6.5. As different ANNs are trained differently, their opinion will also be different even for the same execution sample.

7.4.4.2 Execution of Ensembler Network

Mean (M_r) and variance (V_r) of these opinions (O_a) is calculated for the ensembler network as:

$$M_r = \frac{1}{N_a}\sum_{i=1}^{N_a} O_a^i, \qquad V_r = \frac{1}{N_a}\sqrt{\sum_{i=1}^{N_a}(O_a^i - M_r)^2} \tag{7.5}$$

Here, V_r gives a measure of overtraining of the ANNs under that ensembler even without using testing samples.

7.4.4.3 Execution of Validation Network

Information on validation network will be created by fusing ensembler level information. In the execution phase of validation network, the weighted average validation error (W_V), mean (M_r) and variance (V_r) of ensembler network are used. Opinion of validation network (M_l) is created by the mean of ensembler network (M_r) and their corresponding weighted average validation error (W_V) as:

$$M_l = \sum_{k=1}^{N_r} M_r^k \; W_V^k \tag{7.6}$$

Where, N_r is the number of ensembler networks under the validation network. Similarly, variance of validation network (V_l) is created from mean (M_r) and variance (V_r) of ensembler network with their weighted average validation error (W_V) as:

$$V_l = \sum_{k=1}^{N_r} W_V^k \quad Max\left[(M_r^k - V_r^k - M_l)^2, \quad (M_r^k + V_r^k - M_l)^2\right] \tag{7.7}$$

Distance, D of the validation network is computed as per the Equation-(7.1) shown in Section-7.3.1. If the opinion of validation network is with in the expertise zone of inherited expert, and variance is with in their accepted limit, this validation network is considered as active validation network.

As dimension reduction algorithm is different between different validation networks, weightages (W_d) for dimension reduction can be utilize to give more importance for better dimension reductional algorithm.

7.4.4.4 Execution of Expert Network

Opinion (M_l), variance (V_l) and distance (D) of every validation network under every expert are computed. For execution in the expert level, information of validation networks are fused. First, the distance (D) for every validation networks are calculated and tagged as active validation network if the distance is zero as per Section-7.3.1. Then the number of active validation networks is counted under every expert.

7.4.4.5 Active Expert Network

Active experts are determined depending upon which expert has maximum number of active validation network. If only one expert is active expert, weighted opinion and weighted variance of all active validation networks under this expert are calculated, and considered as the opinion (M_c) and variance (V_c) of the HNN. Where opinion (M_c) of the HNN is formulated as:

$$M_c = \left[\sum_{[j \in N_l]} M_l^j \; W_d^j\right] \left[\sum_{[j \in N_l]} W_d^j\right]^{-1} \tag{7.8}$$

and variance (V_c) of the HNN is formulated as:

$$V_c = \sqrt{\sum_{[j \in N_l]} \left(W_d^j\right)^2 \quad Max\left[(M_l^j - V_l^j - M_c)^2, \quad (M_l^j + V_l^j - M_c)^2\right]} \left[\sum_{[j \in N_l]} W_d^j\right]^{-1} \tag{7.9}$$

where, N_l is the set of active validation network under the same expert and W_d^j is the weightage of the j^{th} dimensional reductional algorithm.

If more than one active expert is available, first, Equation-(7.8) and Equation-(7.9) are used to get individual expert opinion (M_x) and variance (V_x), and then committee machine is used to remove the noisy experts as per Section-7.3 to get the opinion (M_c) and confidence (V_c) of HNN.

7.4.5 Numerical Study of Five Stage Hierarchical ANN

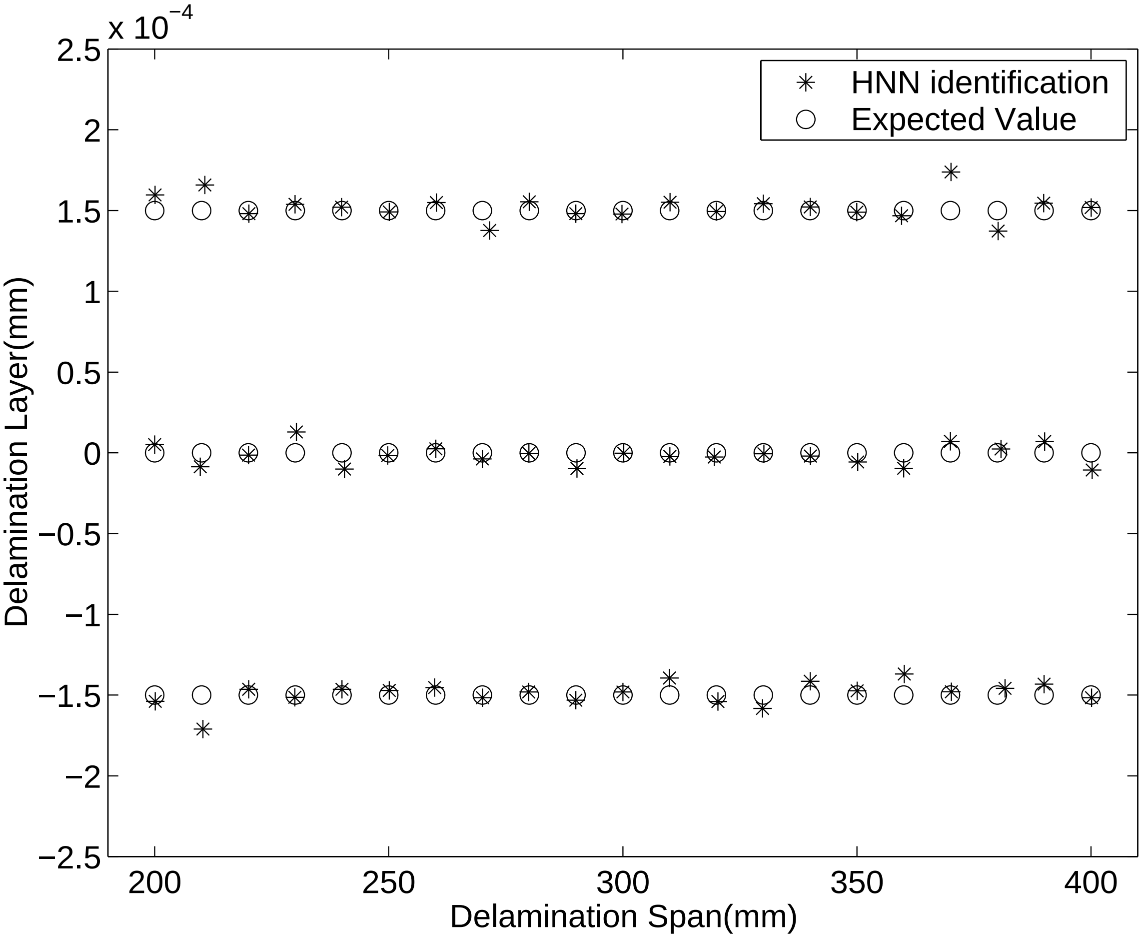


Figure 7.13: Performance of HNN.

In this study, 5 kHz sinusoidal current is considered in the actuation coil as per earlier section. The vibration responses of 551 different cases are numerically simulated. These 551 cases include the intact laminate, laminates with delamination damage at different layers and of 50 different delamination sizes (10 mm to 500 mm) at each layer. Vibration responses of higher dimension for a given delamination, is preprocessed for dimension reduction in the input space of the neural network. Here for simplicity first ten peak values and their location of the sensor open circuit voltage and their time integrals are taken as the input space of the four type of validation neural network. Every validation network consists of four ensemble networks. These ensemble networks are trained and tested by same sample data set but with different random partitioning between training and testing data sets. Hence, every ensemble network is trained and tested by a fixed set of training and testing data sets respectively. In order to identify the delamination length and layer, a simple ANN neural network with 10 inputs and 2 outputs is designed. One hidden layer of node strength 5 is considered in the network. The architecture of the neural network is 10-5-2 and shown in Figure-7.6.

Output space is divided with nineteen span wise overlapped zone as discussed earlier Section-7.3. Thus there are nineteen experts to predict the size and location of the delamination. Every expert is trained by the sample data within their expertise locations. These samples are for the delamination in the corresponding location. A number of delamination identification is performed using proposed Hierarchical neural network and shown in the Figure-7.13, where delamination layers as well as size of the delamination is predicted efficiently.

7.5 Summary

The present chapter has shown the feasibility of using time domain magnetostrictive sensor response for damage detection in composite beam numerically, with the mapping features of the artificial neural networks (ANN). First, a single neural network is used to identify the size of the delamination, where the layer of the delamination is known. Next, the committee machine is used to coordinate different networks, which are exclusively trained for different overlapped zones in the composite beam to identify both the layer and size of the delamination simultaneously. And finally, one five stage hierarchical neural network is trained to systematically consider different issues for mapping with ANN, like overtraining, loss of output uniqueness and damage signature due to dimension reduction of the sensor response and level of confidence in the prediction of the network. The present numerical examples prove that using the artificial neural network, identification of size and location of delamination is feasible.