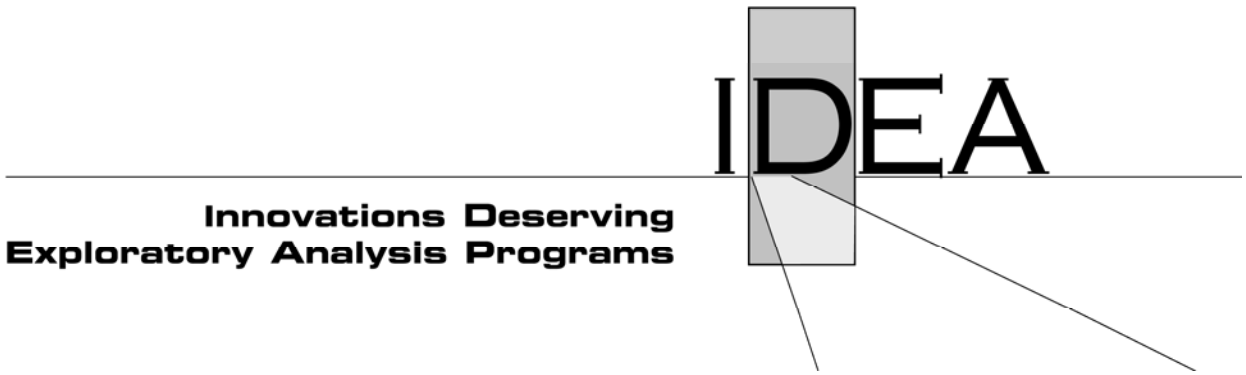


High-Speed Rail IDEA Program

Neural Network Based Rail Flaw Detection Using Unprocessed Ultrasonic Data

Final Report for High-Speed Rail IDEA Project 25

Prepared by:

Jamshid Ghaboussi
University of Illinois at Urbana--Champaign

June 2003

TRANSPORTATION RESEARCH BOARD
OF THE NATIONAL ACADEMIES

INNOVATIONS DESERVING EXPLORATORY ANALYSIS (IDEA) PROGRAMS MANAGED BY

THE TRANSPORTATION RESEARCH BOARD

This investigation by University of Illinois at Urbana--Champaign was performed as part of the High-Speed Rail IDEA program, which fosters innovative methods and technology in support of the Federal Railroad Administration's (FRA) next-generation high-speed rail technology development program.

The High-Speed Rail IDEA program is one of four IDEA programs managed by TRB. The other IDEA programs are listed below.

- NCHRP Highway IDEA, which focuses on advances in the design, construction, safety, and maintenance of highway systems, is part of the National Cooperative Highway Research Program.
- Transit IDEA focuses on development and testing of innovative concepts and methods for improving transit practice. The Transit IDEA Program is part of the Transit Cooperative Research Program, a cooperative effort of the Federal Transit Administration (FTA), the Transportation Research Board (TRB) and the Transit Development Corporation, a nonprofit educational and research organization of the American Public Transportation Association. The program is funded by the FTA and is managed by TRB.
- Safety IDEA focuses on innovative approaches to improving motor carrier, railroad, and highway safety. The program is supported by the Federal Motor Carrier Safety Administration and the FRA.

Management of the four IDEA programs is integrated to promote the development and testing of nontraditional and innovative concepts, methods, and technologies for surface transportation.

For information on the IDEA programs, contact the IDEA programs office by telephone (202-334-2065); by fax (202-334-3471); or on the Internet at http://www.nationalacademies.org/trb/idea

IDEA Programs
Transportation Research Board
500 Fifth Street, NW
Washington, DC 20001

Neural Network Based Rail Flaw Detection Using Unprocessed Ultrasonic Data

IDEA Program Final Report
for the Period October 2000 Through April 2003
Contract Number NAS 101, Task order No. 3, HSR-25,
with Modifications No. 1 and 2

Prepared for
the IDEA Program
Transportation Research Board
National Research Council

Jamshid Ghaboussi

Department of Civil and Environmental Engineering
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

Submittal Date:
June 2003

Neural Network Based Rail Flaw Detection
Using Unprocessed Ultrasonic Data

Jamshid Ghaboussi

Table of Contents

Neural Network Based Rail Flaw Detection Using Unprocessed Ultrasonic Data

1. Executive Summary

In the current practice in rail flaw detection the raw (unprocessed) data gets processed to generate the data for the simple visual displays that can be produced for the operator. In the past Sperry Rail Service has funded research on application of neural networks in the rail flaw detection. All the previous research has been based on using processed ultrasonic data.

Useful information gets discarded in processing the ultrasonic data. This research project was intended for developing methods that can use the unprocessed ultrasonic data in railroad rail flaw detection. The reasoning behind this project was that by directly using the unprocessed data with appropriately designed and trained neural networks it would be possible to make maximum use of the information in the ultrasonic data to improve the efficiency and reliability of rail flaw detection. Sperry Rail Service was to be the co-funder and participant in this project.

The research project was to be carried out in several stages. It was planned that we would start the project by establishing an ultrasonic rail flaw detection laboratory at the campus of University of Illinois at Urbana-Champaign with loaned equipment from Sperry Rail Service. Simultaneously, methods would be developed for collecting, digitizing and storing the unprocessed ultrasonic data. The volume of the unprocessed data is very large and currently it is not being stored; only the processed data gets stored. The next stage was collecting the ultrasonic data in the laboratory and at Sperry's test track at Danbury, CT. The collected data would have been used in designing, developing and training a set of rail flaw detection neural networks. These neural networks would have gone through several rigorous cycles of testing, evaluation and retraining.

After an initial study, it was decided to establish the laboratory at Danbury. The scope of the project was changed and it was decided that the unprocessed data would be collected only

in the laboratory under conditions similar to those in the field, and by using the same transducer sets. The necessary equipment was acquired and the laboratory was established. A small sample of data was collected. At this point Sperry Rail Service had to withdraw from the project for internal reasons, and without their participation it was not possible to complete the project. Therefore the project was terminated.

2. The statement of the problem

The purpose of this project was to develop and test neural network based methods to improve the reliability and speed of ultrasonic railroad rail inspection and rail flaw detection, to enable earlier detection of flaws and to detect certain heretofore undetectable flaws. The current rail flaw detection technology is limited by the human operator's ability to interpret the ultrasonic data stream. This limits the detection car operating speeds and allows some important and critical flaws to go undetected. The results from an ongoing project co-funded by Sperry Rail Service and AAR, showed that neural networks can improve the rate and reliability of rail flaw detection by using the same processed data that is used in the operator-based system. Further improvements are possible by using the unprocessed data, which contains more information than the processed data.

Development and application of the proposed technology is of particular importance on lines that, in the future, will combine the high speed passenger rail operations with freight traffic and heavier axle loads will further complicate the situation. They are likely to contribute to a higher rate of initiation and growth of rail flaws. Failure to detect and repair them in a timely fashion could cause service reliability problems and will pose safety concerns as well.

Improved rail flaw detection resulting from this project would to lead to more reliable, earlier detection of smaller flaws, before they grow and become critical. This research is also likely to lead to more reliable methods to determine the size of the flaws. Smaller flaws often do not pose as great a hazard as larger flaws, and as such do not need to be repaired immediately.

Higher reliability in detection of all sizes of flaws and determination of the flaw sizes will facilitate implementation of more efficient management of rail repair.

2.1. Ultrasonic railroad rail inspection

Railroad rails are routinely inspected by electro-magnetic induction and/or ultrasonic methods to detect flaws and to identify their type. The operator in a detection car inspects the railroad rails using processed ultrasonic data. This project was co-funded by Sperry Rail Service and it was intended to use the data generated by Sperry in the laboratory and in the field using their ultrasonic transducers. In the following we briefly describe the Sperry rail inspection cars.

A Sperry Rail Service road/rail detection car is shown in Figure 1. These detection cars typically have an ultrasonic inspection unit trailing the rear wheels, as seen in Figure 1. The ultrasonic transducers are installed in two wheels over each rail, as shown in Figure 2. The pliable wheels are filled with a coupling fluid and they are in contact with the rails under pressure. The transducers are arranged to send ultrasonic signals at different angles into the rail, specially the rail head. The stream of signals are processed and gated, and the results are displayed in strip chart format on a monitor in front of the operator. The ultrasonic strip chart is constructed from a stream of records and each record contains 16 bits of binary data, which includes the processed signals generated by all the transducers.

The ultrasonic test data used in training of the neural networks in earlier studies was generated by inspection runs over the Sperry test track which contains a number of known defects. The location and type of the defects was determined from Sperry Rail Service's test track defect manual. The strip chart data contained within a window of prescribed size were used as the input to the neural networks. The window size refers to the number of consecutive records included in a neural network input vector. The window distance is the distance between the centerline of two adjacent windows. The neural network input vector is generated according to the window size with the centerline on the defect location, as shown in Figure 3. Moreover, as shown in Figure 4, if a defect is extended over a section of the rail longer than the window size, a sequence

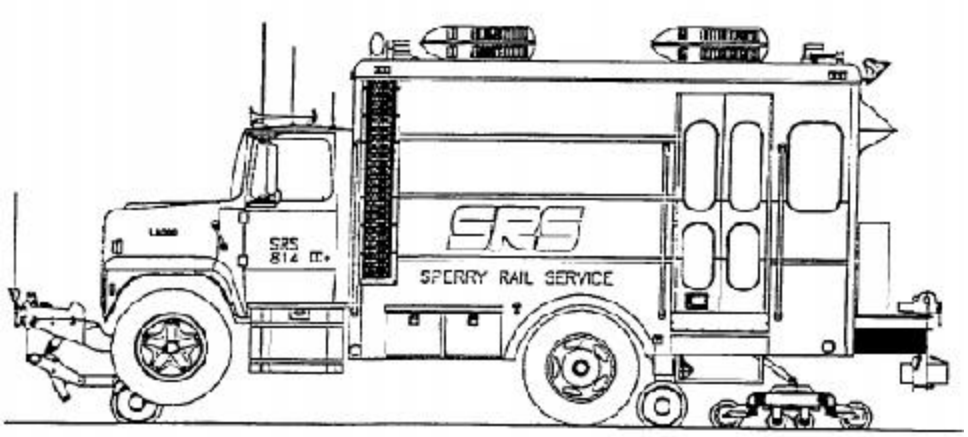


Figure 1. A road/rail ultrasonic detection car.

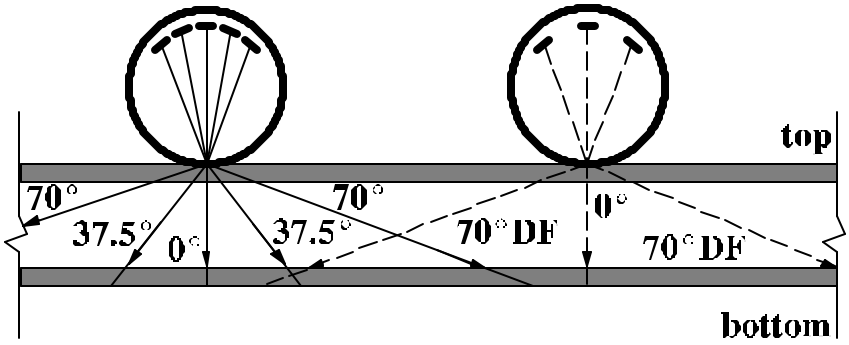


Figure 2. The ultrasonic transducers.

of neural network input vectors is generated from windows separated by window distance. Finally, the same procedure is also used to generate sequences of neural network input vectors for clean rails without any defect, as shown in Figure 5. Throughout the earlier studies , we have used a window size of 7 records and a window distance of 12 records. With the window size of 7 records and each record containing 16 binary bits, each neural network input vector contains 112 binary bits.

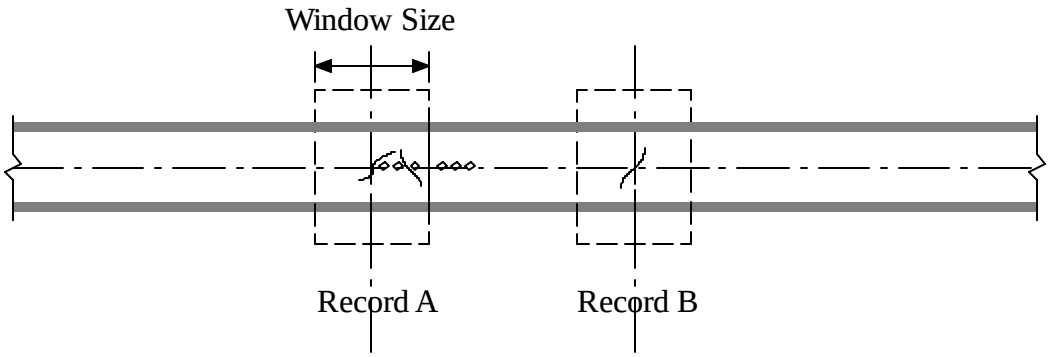


Figure 3. Damage occurs at a single point.

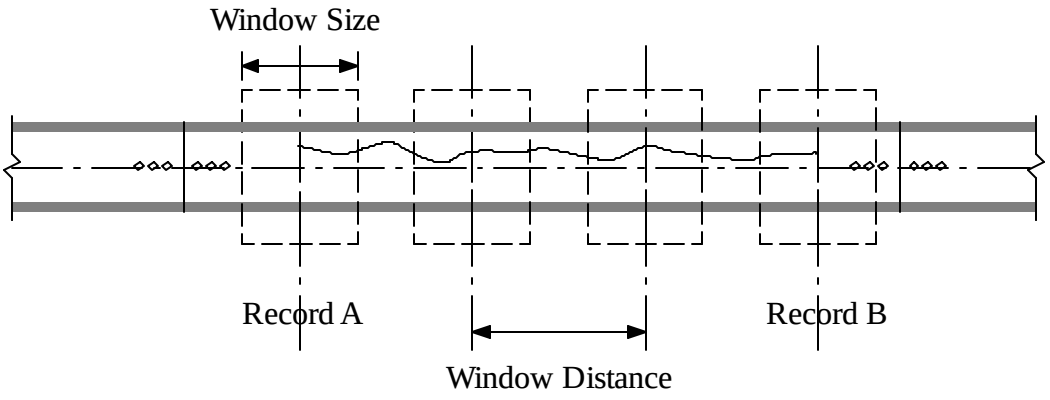


Figure 4. Damage occurs in a range between record A and record B.

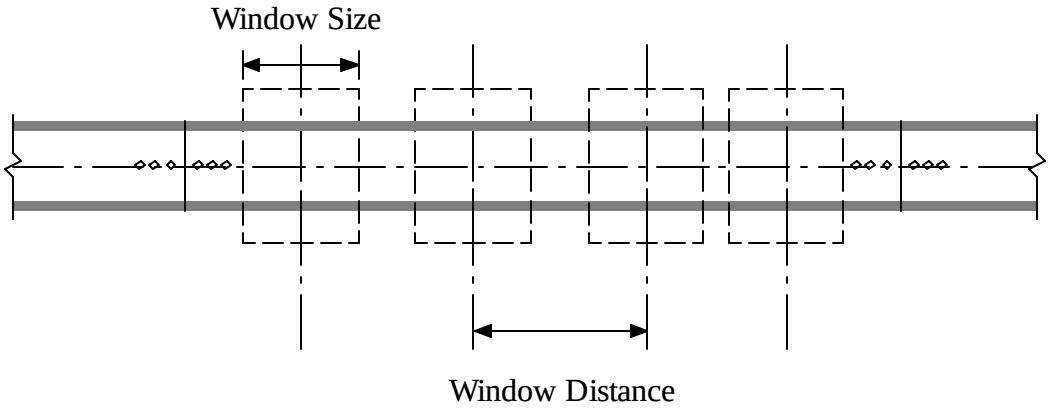


Figure 5. Clean Rail.

In the initial phase of the earlier studies the same processed data that the operator sees was used in the neural network study. The intention was that the successful development and implementation of neural network-based flaw detection techniques will assist the operators and will improve the reliability and efficiency of railroad rail flaw detection.

2.2. Neural networks

Artificial neurons

Artificial neural networks are constructed as an assemblage of artificial neurons that are roughly modeled after the biological neurons in the brains and nervous system of humans and animals. We present a brief and simplified introduction to the structure and operation of the the biological neurons.

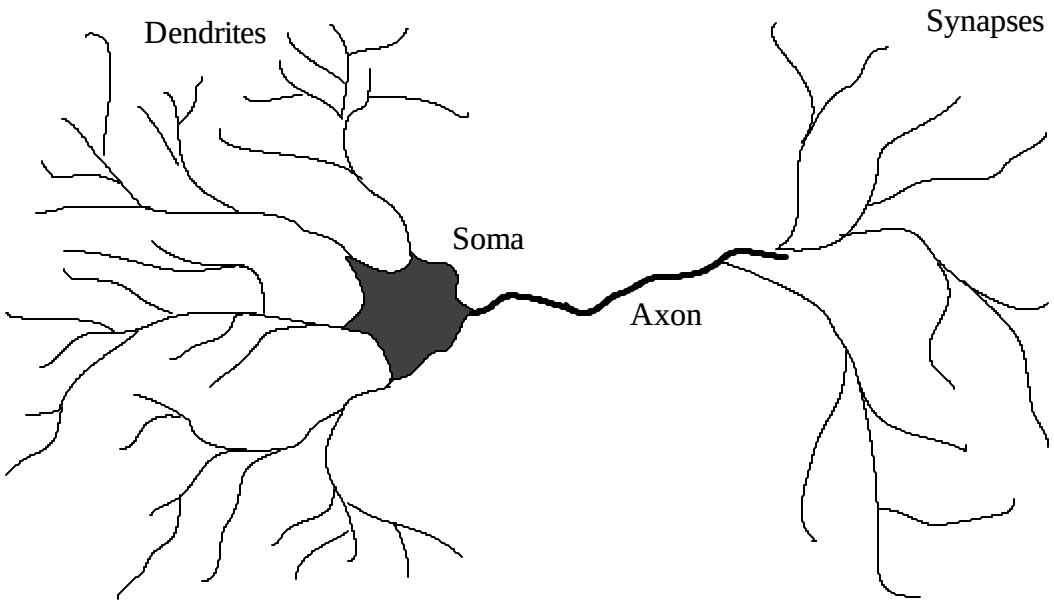


Figure 6. A simplified schematic representation of a biological neuron.

Each biological neuron is connected to a large number of other neurons. Electrical signals travel along these connection. These signals arrive at the neurons along the connection called

the “dendrites.” These signals produce a physio-chemical reaction in the main body of the neuron called the “soma,” which may result in generation of an electrical charge. The electrical charge causes a signal to travel along the “axon” and to be transmitted to the neurons along the “synoptic” connections. Figure 6 schematically shows the main elements of a biological neuron.

Neural networks are composed of a number of interconnected artificial neurons. A vast majority of the artificial neurons used in the current generation of neural networks are based on the model proposed by McCulloch and Pitts in the 1940’s. The McCulloch-Pitts artificial neuron was binary.

An artificial neuron is shown in Figure 7. Shown on the left hand side of this Figure are a number of incoming connections, transmitting the signals from the other artificial neurons. A numerical value, called the connection weight, is assigned to each connection to represent its effectiveness or its strength in transmitting the signals. The weight of the connection from node number j into node number i is w_{ij}, and the signal coming from the node number j is S_j. The incoming connections are modelling the dendrites in the biological neurons.

The artificial neuron itself represents the soma in its biological counterpart. The physio-chemical reactions that take place within the soma and cause it to fire a signal are represented by two simple operations shown in the two circles. The first operation is the weighted sum of all the incoming signals, each weighted by the weight of the connection on which it is travelling.

$$z_i(n+1) = \sum_j w_{ij} S_j(n) - \Theta_i$$

In this equation Θ_i is the bias of the neuron. In reality, the operation of the artificial neuron is not affected by the magnitude of the time step.

The second operation within the artificial neuron consists of passing the results of the weighted sum through an “activation function”, f (x). The result of this operation is called the

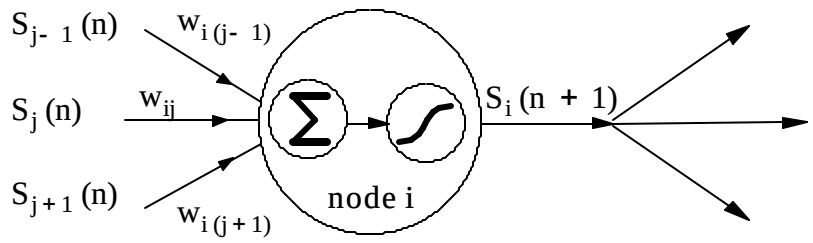


Figure 7. An artificial neuron.

"activation" of the neuron and it is denoted by $S_i(n+1)$. Activation functions are usually bounded functions varying between zero and one, and they provide the main source of nonlinearity in neural networks.

Real valued neurons, that are widely used, have activations values in the range of [0, 1] or [-1, 1]. The most commonly used activation function is the sigmoid function given in the following equation.

$$S_i(n+1) = f[z_i(n+1)] = \frac{1}{1 + e^{-l z_i(n+1)}}$$

The sigmoid function is a smoothed version of the binary step function and similar to the step function it varies between 0 and 1. However, the transition is more gradual and it has a real non-zero value for all the possible values of its argument.

Another common choice for the activation function is the hyperbolic tangent function that is a bounded function varying between -1 and 1.

$$f(x) = \tanh(a x)$$

Multi-layer feedforward neural networks

Multi-layer Feedforward (MLF) neural networks are probably the most widely used neural networks. With a few exceptions the vast majority of the neural applications in engineering applications use the MLF neural networks. Unlike the randomly connected or the fully connected Hopfield nets, the MLF neural networks are not dynamical systems and consequently, they least resemble the nervous system in humans and animals.

The artificial neurons in the MLF are arranged in a number of layers. The first layer is the input layer and the last layer is the output layer. The layers between the input and the output layers are referred to as the hidden layers. The order of the layers and the direction of the propagation of the signals is from the input layer, through the hidden layers to the output layer. In the fully connected version, each node is connected to all the nodes in the next layer. Figure 8 shows a typical MLF neural network.

The nodes in the input layer are not quite artificial neurons. They only receive the input values and transmit them to the artificial neurons in the first layer which is usually the first hidden layer.

The type of fully connected neural network shown in Figure 8 is the most commonly used. However other patterns of connections are also possible. Some patterns of connectivity can be the result of adaptive architecture determination.

The nodes in the MLF neural networks are the typical artificial neurons that were described in an earlier section. The activation of the nodes are determined from an activation function and a weighted sum operation.

$$z_i^k = \sum_j w_{ij}^k \, S_j^{k-1} - \theta_i$$

$$S_i^k = f[z_i^k]$$

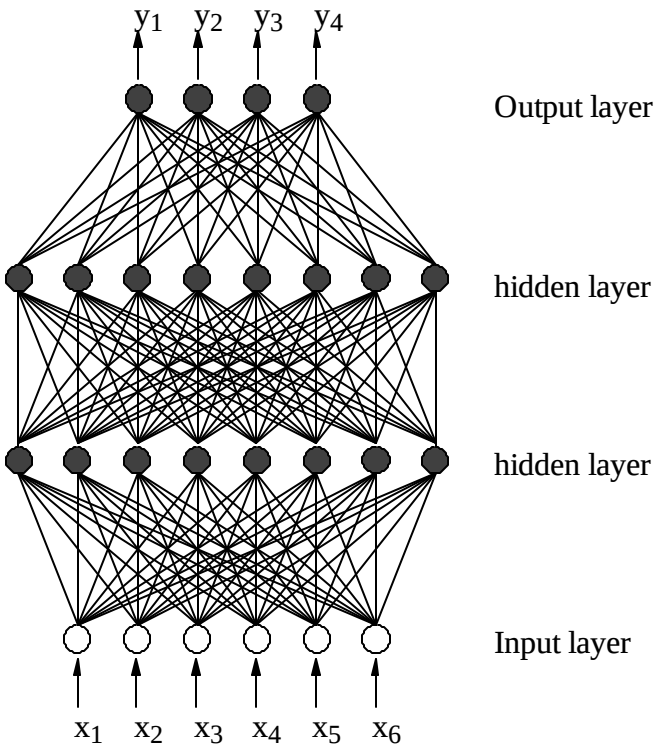


Figure 8. A multi-layer feed-forward neural network.

The superscript k is used to designate the layer number which varies from zero for the input layer to n for the output layer. In the equation θ_i is the bias, w_{ij}^k are the weights the connections coming into the layer number k, and S_i^k is the activation of node number i in layer number k,. The input vector can be considered as the activations of the input nodes and the activation of the output nodes are the output of the neural network.

$$S_i^0 = x_i$$

$$y_i = S_i^n$$

The activation function for the nodes is a bounded function varying between 0 and 1 or between -1 and 1. In binary neural networks the activation function is a step function. In the real-valued neural networks the activation function is either a sigmoid $f(x) = 1/(1 + e^{-lx})$, or hyperbolic tangent $f(x) = \tanh(ax)$.

How many hidden layers are needed

To start with there are no rigorous general rules for determining the appropriate number of hidden layers. Like many aspects of neural networks, the number of hidden layers is problem dependent. The author's own experience, as well as a general consensus among the users of neural networks, is that no more than two hidden layers is needed for a vast majority of problems. As the number of hidden layers increase beyond two, the correlation between the input layer and the output layer diminishes and the training of the neural network becomes more difficult.

The question of whether one or two hidden layers are needed depends to some extent on the nonlinearity and the complexity of the underlying association in the training data that the neural network is expected to learn. One hidden layer is sufficient for many problems. If the problem can be solved and the neural network can be trained with one hidden layer, then it is preferable not to use two hidden layers for that problem. However, for many practical problems one hidden layer is not sufficient.

The vast majority of neural networks in engineering applications use two hidden layers and most of these problems can not be solved with one hidden layer. This is because of the high degree of nonlinearity in most of the engineering problems.

Of course, there are some exceptions to the rule of a maximum of two hidden layers. There are some cases, like the replicator neural networks which may require three hidden layers. In some applications a composite neural network may appear to have up to four hidden layers. However, these neural networks are composed of more that one neural network, and the constituent neural networks are trained separately.

Training of MLF neural networks

The response (output) of a MLF neural network to any given stimuli (input) obviously will depend on the connection weights. The choice of the activation function also has an influence on the stimulus-response behavior of neural network. However, the activation function is a fixed part of the neural networks and it does not change during the training of the neural network. The training of a neural network essentially means the adaptation of the connection weights.

The training of the MLF neural networks is termed "supervised learning" since the neural network learns from the patterns of input-output pairs. The knowledge to be learned and acquired by the neural network is contained in the set of input-output patterns that constitutes the training data set as shown in the following equation.

$$[\,\mathbf{Y}_1,\ \mathbf{X}_1\,],\ \cdots,\ [\,\mathbf{Y}_k,\ \mathbf{X}_k\,]$$

During the training the connection weights of the neural network are changed so that for each input vector $\mathbf{X}_i$ the error at the output between the computed and desired output vector $\mathbf{Y}_i$ is minimized. The output error is defined as follows.

$$e_p = \frac{1}{2} \left\| \overline{\mathbf{Y}}_p - \mathbf{Y}_p \right\|_2^2 = \frac{1}{2} \sum_{i=1}^{M} (\overline{y}_{pi} - y_{pi})^2$$

The total error E is the sum of the errors for all the input-output pairs in the training data set.

$$E = \sum_p e_p$$

Obviously, the total error in the output of the neural network is a function of its connection weights.

$$E = E(w_{ij})$$

The essence of the training of a neural network is to determine a set of connection weights that minimize the total error E. The rules used to update the connection weights is called the learning rule.

Almost any method of optimization can be used to determine the optimal connection weights. The most commonly used method is the iterative method of updating the connection weights based on a simple variation of the gradient descent method.

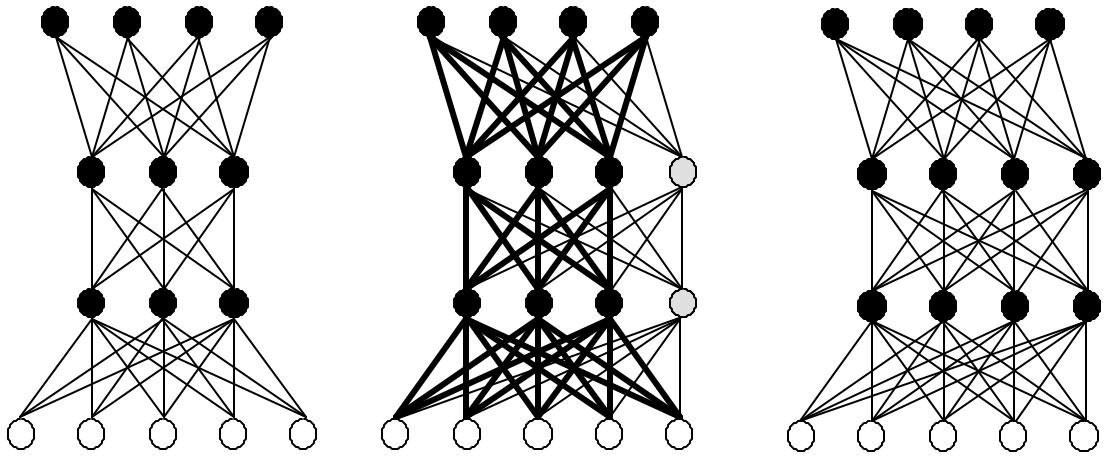
$$\Delta w_{ij} = - \eta \frac{\partial E(w_{ij})}{\partial w_{ij}}$$

In this equation η is the "learning rate". It is usually a small number between 0 and 1. Learning rate is an important parameter which governs the rate of convergence of the gradient based algorithm.

Adaptive architecture

When the neural network is used to solve a problem, it is important to decide the optimal architecture of the network. In order to obtain good generalization capability, one has to build into the network as much knowledge about the problem as possible, and limit the number of connections appropriately. Therefore, it is desirable to find algorithms that not only optimize the weights for a given architecture, but also optimize the architecture itself. This means in particular optimizing the number of layers and the number of neurons per layers.

There are several methods to construct the optimal architecture, such as dynamic node creation, the cascade-correlation learning architecture, skeletonization, pruning, and dynamic hidden elements generation. Basically, there are only two major algorithms: network growing and network pruning. For network growing algorithms, the network begins with a basic one, and neurons are added during the training. The network is easy to extend as new patterns are added to learn. In addition, such a network freezes the original trained weights and adjusts the new



(a) (b) (c)

Figure 9: The adaptive method of neural network architecture determination.

weights to new learning patterns. On the other hand, for network pruning algorithms, the network begins with a large one, and the redundant neurons and connections are pruned during the training. The disadvantage of such a network is that the old network can not be used and has to be trained over again when new learning patterns are added. Therefore, the network growing algorithm is preferable to the pruning for the class of problems considered here.

The author and his co-workers have proposed an adaptive method of architecture determination, which generates new hidden neurons dynamically. In Figure 9(a), the network is started with a small number of neurons at the hidden layers. In Figure 9(b), an additional neuron is added to each hidden layer at a time when the criterion of adding new nodes are encountered. The criterion is defined according to the learning performance of the current network. In Figure 9(c), when a hidden node is added to the hidden layer, connection weights of this new node to all the other nodes are created and initialized. For the new connection weights to acquire the portion of the knowledge which has not been learned by the old connection weights, some training is performed only for the new connection weights while the old connection weights are frozen. Then the training continues for all the connection weights. These steps will be repeated

and new nodes are added to the hidden layers as needed until the present network satisfies the convergence criterion. At the end of training, the appropriate network architecture is determined automatically.

2.3. Application of neural networks in the ultrasonic rail flaw detection

The main concept behind the application of neural networks in ultrasonic rail inspection is shown in Figure 10.

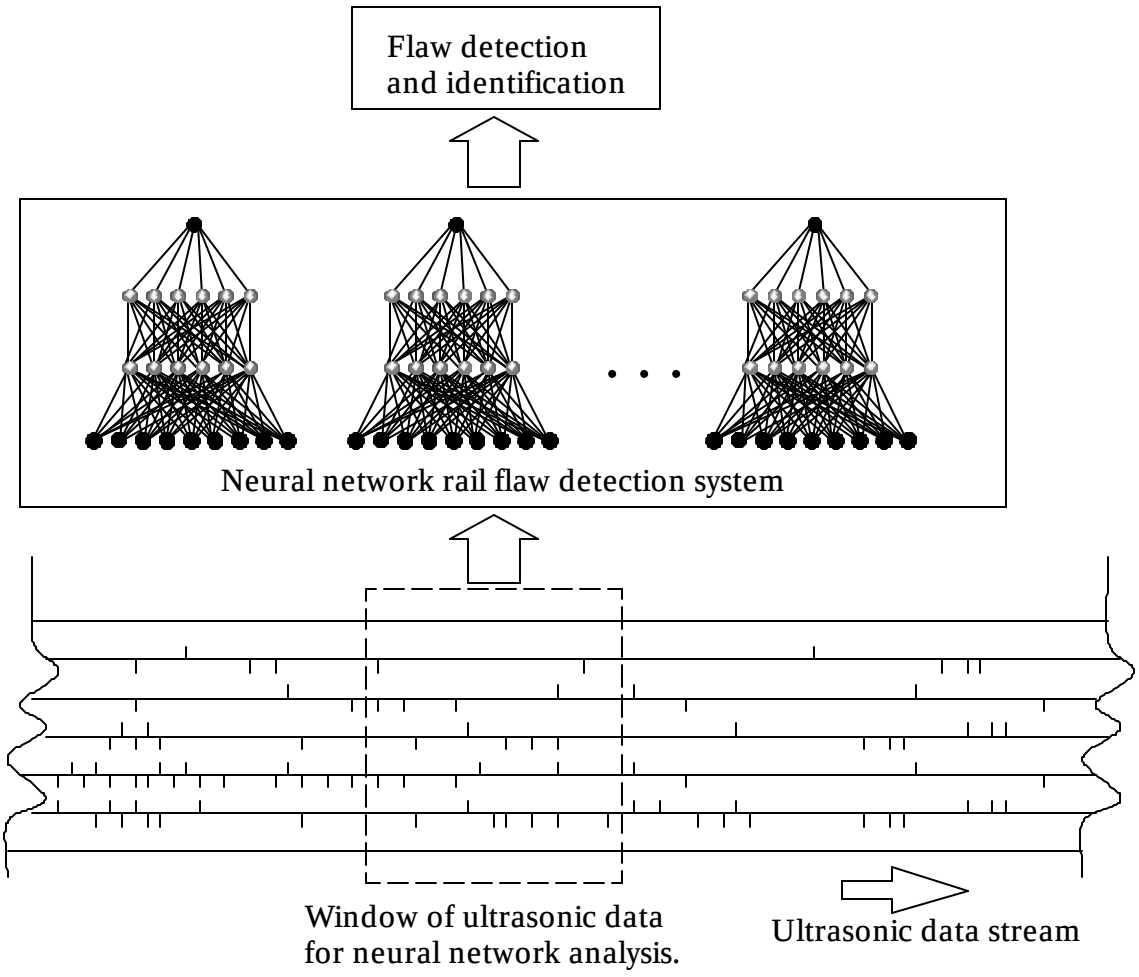


Figure 10. Schematics of ultrasonic flaw detection with trained neural networks.

As the inspection car travels over the rails the ultrasonic inspection is being performed continuously. The specially designed wheels that contain the ultrasonic transducers are in contact with the rails. The transducers are sending and receiving ultrasonic signals The ultrasonic signals received by the transducers go though signal processing and the processed data are displayed on the monitors in front of the operator. Figure 10 shows a stream of strip chart data. The operator makes a decision from the processed data on the existence of a flaw.

The same task can also be performed by the a set of trained neural networks. As shown in Figure 10, these neural networks are designed to receive the ultrasonic data within a moving window. A number of studies in the early stages of the project funded by Sperry Rail Service determined the appropriate size of the window. The data from the moving window is passed through the neural network or a set of neural networks. The output of the neural networks indicate the existence of flaws. The neural networks can also be trained to provide information about the type of the flaw.

These neural networks have to be trained with an appropriate training data set. The training data is collected from the normal operation of the detection car and the processed data that they generate. Since the neural networks obtain all their information from the training data, special precautions must be taken to assure that training data contains all the information that neural networks need to be as effective as possible in the detection of flaws.

The first task is to determine the architecture of the neural network. Next the appropriate training data set is collected and used to train the neural networks. Then the trained neural networks are tested with new data that was not used in the training. The process of training and testing is repeated through several cycles, until a satisfactorily trained neural network is arrived at. However, the training of the neural network never completely stops. If new data becomes available to increase the effectiveness of the neural network, they can always be retrained to acquire the additional information in the new data set.

2.4. Processed ultrasonic data

All the ultrasonic railroad rail inspection is currently done with the processed data. The signals received by the ultrasonic transducer get processed to generate simple visual displays that can be used by the operator. There are two basic types of processed data.

The strip chart method is the simple processing method that has been in use for years, even before microprocessors and personal computers. In recent years, the strip chart data is generated digitally and displayed on monitors in front of the operator. The processed strip chart data is in binary from; when a returned ultrasonic signal exceeds a threshold a positive signal is generated. For each channel the strip chart data appears in the form of ticks, as shown in Figure 10. Each line represents one ultrasonic transducer. When a tick appears, it is an indication that a returned signal exceeding the threshold has been received by that transducer. There are two sets of lines for the two rails, each set of lines represent the transducers over one rail.

The B-scan processing method generates more information for the operator than the strip chart method. In the B-scan processing method when a return signal exceeds a threshold the distance from the object that caused the reflection is also determined from the time of arrival of the return signal. For each transducer the direction of propagation of the ultrasonic signal is known. When the distance is also known, the location of the object generating the return signal can be determined. Collection of successive locations make up two-dimensional images detected by the transducers. The operator sees a collection of two-dimensional images displayed on the monitor.

2.5. Unprocessed ultrasonic data

Processing of the ultrasonic data, either by the strip chart method or by the B-scan method, was intended to generate visual data that can be quickly processed by the operator. Human operators can only process a limited amount of information at the operating speed of the detection car. The volume of the unprocessed data is very large and it can not be displayed for the operator. However, a computerized method such as trained neural networks can process very large volume

of data at high speeds. The computerized methods are only limited by the speed of the processing computer.

The advantage of using the unprocessed data is that it contains a lot of information that is lost during the standard processing methods. Potentially this additional information can be used to increase the detection rates and to enable the detection of flaws that are difficult to detect at the present.

3. Previous research

All the previous research on application of neural networks in rail flaw detection that was conducted at University of Illinois at Urbana-Champaign was funded by Sperry Rail Service. All this research was done on Sperry equipment. A brief outline of this research is presented in the in the following section.

3.1. Neural networks in strip chart method

For the OMNI+DF system, there are 16 bits in each record of the ultrasonic strip chart data. They are the first 16 bits in Figure 11. For UX9+VSH system, there are two more VSH bits in each record, which are the last two bits shown in Figure 11.

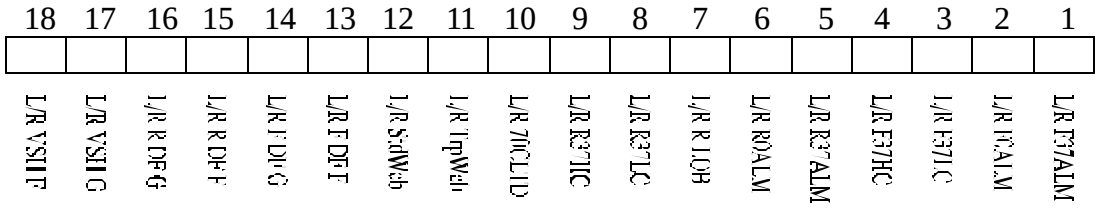


Figure 11. A 18-bit signal record

A window size of seven records was chosen for the neural network input. Since the distance between the records is 3 inches, each window covers 18 inches of the rails. The size of

the input vector for the neural networks is equal to the window size of seven records times the number of bits per record. This results in a input vector of 112 binary bits in OMNI+DF system and 126 binary bits in UX9+VSH system.

Table 1. Classification of defect types

No.	Type	Defect detection	Defect identification
0	Clean (no defect)	0	0 0 0 0 0 0 0 0 0 0
1	BHJ	1	0 0 0 0 0 0 0 0 0 1
2	HSJ	1	0 0 0 0 0 0 0 0 1 0
3	HWJ	1	0 0 0 0 0 0 0 1 0 0
4	HSH	1	0 0 0 0 0 0 1 0 0 0
5	VSH	1	0 0 0 0 1 0 0 0 0
6	TDD/TDC/TDT/EBF	1	0 0 0 1 0 0 0 0 0
7	Crushed Head	1	0 0 1 0 0 0 0 0 0
8	Extra Drillings	1	0 1 0 0 0 0 0 0 0
9	Torch Cut	1	1 0 0 0 0 0 0 0 0

Later we adopted a new approach by training multiple neural networks for the defect identification. A separate neural network was trained for each defect type. The data would pass through multiple neural networks and each would detect a specific defect. The earlier multiple neural networks had only one output node. In the “delta function” method the output is binary. A defect is detected when the output node is on (output value of 0.9) and the defect is located at the center of the window. Otherwise, the output node is off (output value of 0.1), indicating no defects. In the “linear function” method the value of the output node during the training depends on the location of the defect within the window. The output value varies from 0.3 for the defect at the edge of the window to 0.9 when the defect is at the center of the window. The absence of defect is still indicated by the output value of 0.1. The delta function and the linear function are shown in Figure 12.

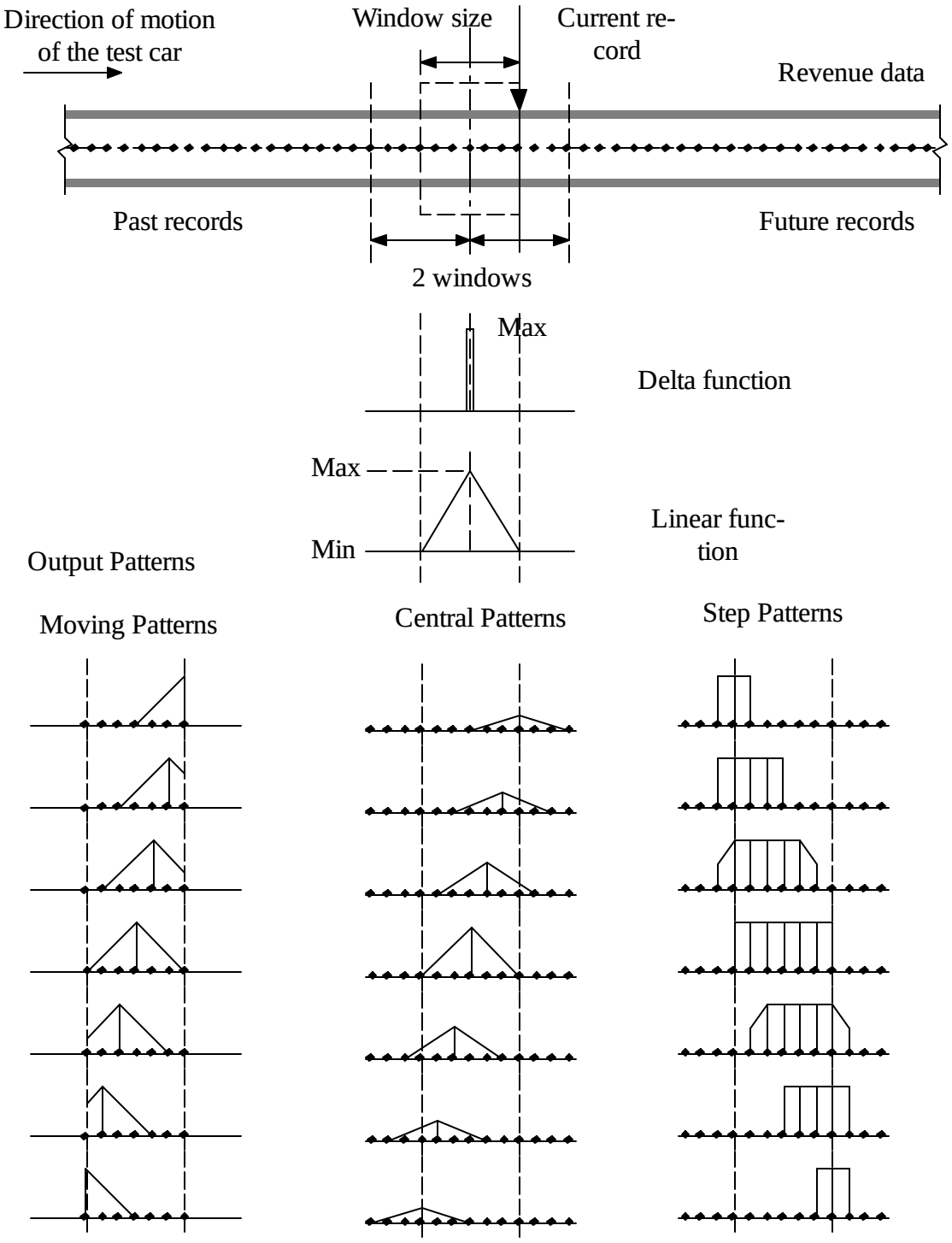


Figure 12. Neural network output patterns

The output of the neural networks indicates the presence or absence of defects and information about the defect type. Early on in this study the defects were classified into ten types, including clean rails, as shown in Table 1. Initially, each defect type was assigned nine binary bits in the neural network output vectors. For practical computational reasons the binary values of "0" and "1" are replaced by "0.1" and "0.9".

At a later stage in the study a new variation was introduced in representing the output of the neural networks with seven nodes. These neural networks were deemed to have improved capability for learning the defect detection. The value of the output node depends on the location of the defect within the window. Three different patterns shown in Figure 12 were tested. For the moving patterns, the center of a linear triangular function with values of 0.3, 0.5, 0.7, 0.9, 0.7, 0.5, 0.3 is located over the defect in the window. The portions of the triangle falling outside the window are truncated. For the central patterns, the center of a linear triangular function also is located over the defect in the window. However, its value depends on the location of the defect. The value of the center of the triangle is 0.9 when the defect is at the center of the window. The values of the triangular pattern are multiplied by 1.0, 0.75, 0.5, 0.25 as the defect moves from the center of the window to its edge. For the step patterns, a step function is symmetrically located over the defect and the width of the step function depends on the location of the defect in the window, as shown in Figure 12.

Data for Training of the Neural Networks

Two methods have been used to prepare the data for neural network training. Initially we prepared the training data according to Sperry's defect manual for the test track. The DF channels were used to determine the rail ends. The record in the middle of the section of the DF signals near the rail end was used to locate the beginning and the end of each rail. The training patterns of rail flaws were decided based on the locations of rail flaws in each rail in the defect manual. The training patterns for clean rails were also generated using the test track data.

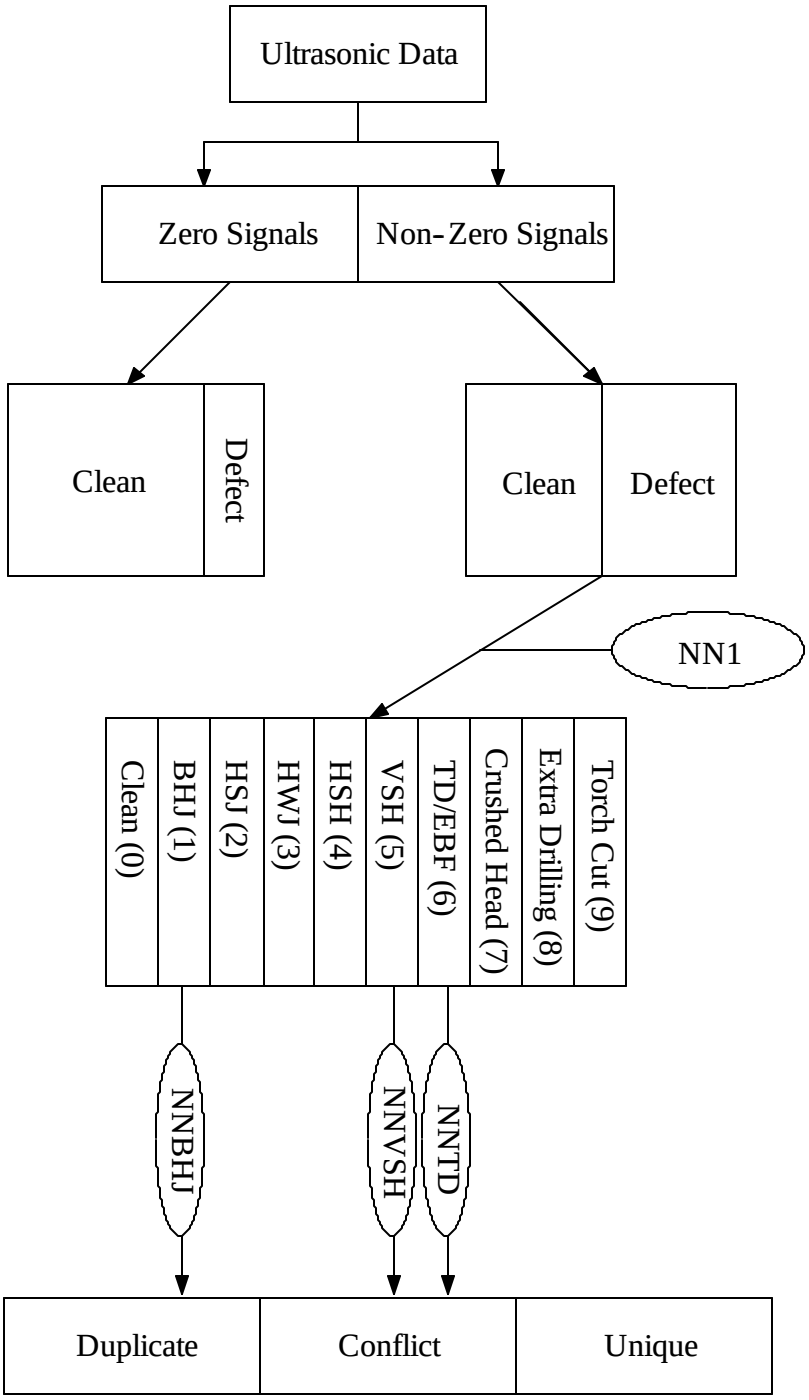


Figure 13. Classification of the data for neural network training

The classification of the data for neural network training is shown in Figure 13. When there are no non-zero signals within the window, it is an indication of clean pattern and absence of any defects. However, sometimes the ultrasonic transducers do not generate any signals for a defect. In either case the windows with all zeros were not included in the training of the neural networks, since they will always produce a a zero output. Only the windows with non-zero signals were used in training the neural networks. These windows were grouped into clean and defect patterns. Three types of patterns occur in the non-zero records: duplicates, conflicts and unique patterns. Duplicate patterns are defined as the group of patterns that have the same input and output vectors. One from each group of duplicate patterns is included in the training of neural networks. The conflicts are defined as the patterns that have the same input vectors but different outputs. All the conflicts were excluded from the training data since the neural networks have no way of learning these patterns. The remaining records were unique and they were all included in the training of the neural networks.

The number of defect patterns is far less than the number of the clean patterns. Consequently, the neural networks could become biased in favor of clean patterns during the training process. In order to prevent this and to make the training process less biased, a multiplication factor was applied to the defect patterns to balance the numbers of defect and clean patterns.

Data for Testing of the Neural Networks

Test track and revenue data with known locations of defects were used to test the capability of the trained neural networks. The data for testing was intentionally not used in the training of the neural networks in order to test their generalization capability.

Two parameters can be used in deciding whether the output of the neural network during the testing process indicates the presence of a defect. The first parameter is a threshold for the output nodes. The second parameter is the number of output nodes (in the multiple output neural networks) that exceed that threshold. For all the output patterns shown in Figure 12, the threshold starts at 0.5. For moving and central patterns, the number of output nodes larger than the

threshold starts at 1. For step patterns, the number of output nodes larger than the threshold starts at 3. Once the number of records in a window greater than or equal to the threshold is larger than the above specified numbers, the central record is then taken as a defect. Otherwise, it is treated as clean. During the training of the neural networks they are tested at regular intervals with a specified threshold and the limit on the number of output nodes exceeding the threshold. The connection weights of the neural networks are also saved at these intervals. At the end of the training process, the connection weights corresponding to the best test results is selected and used for the trained neural network.

Neural Network Architectures

All the neural networks used in this study are multi-layer feed-forward neural networks. Moreover, they all have four layers: input layer, two hidden layers, and output layer. The same four layers were used for both the first level and the second level neural networks. A typical first level neural network is shown in Figure 14. This neural network is shown with a delta function output. The same type of neural network was used in the linear function output. All the first level neural networks have a single output node. A typical second level defect identification neural network is shown in Figure 15. All the second level neural networks have seven output nodes. The same neural network architecture was used for the three output patterns shown in Figure 12, namely, the moving patterns, the central patterns, and the step patterns.

The number of input nodes depends on the window size. For the window size of 7 records, there are 112 (7 ¢ 16) input nodes for the 16-bit OMNI+DF system and 126 (7 ¢ 18) input nodes for the 18-bit UX9+VSH system. The number of the nodes in the hidden layers are determined adaptively during the training.

Training and testing of the neural networks

Sets of neural networks were trained and tested initially with the test track data. These neural networks were subsequently tested with revenue data. Extensive studies were performed and the trained neural networks were tested.

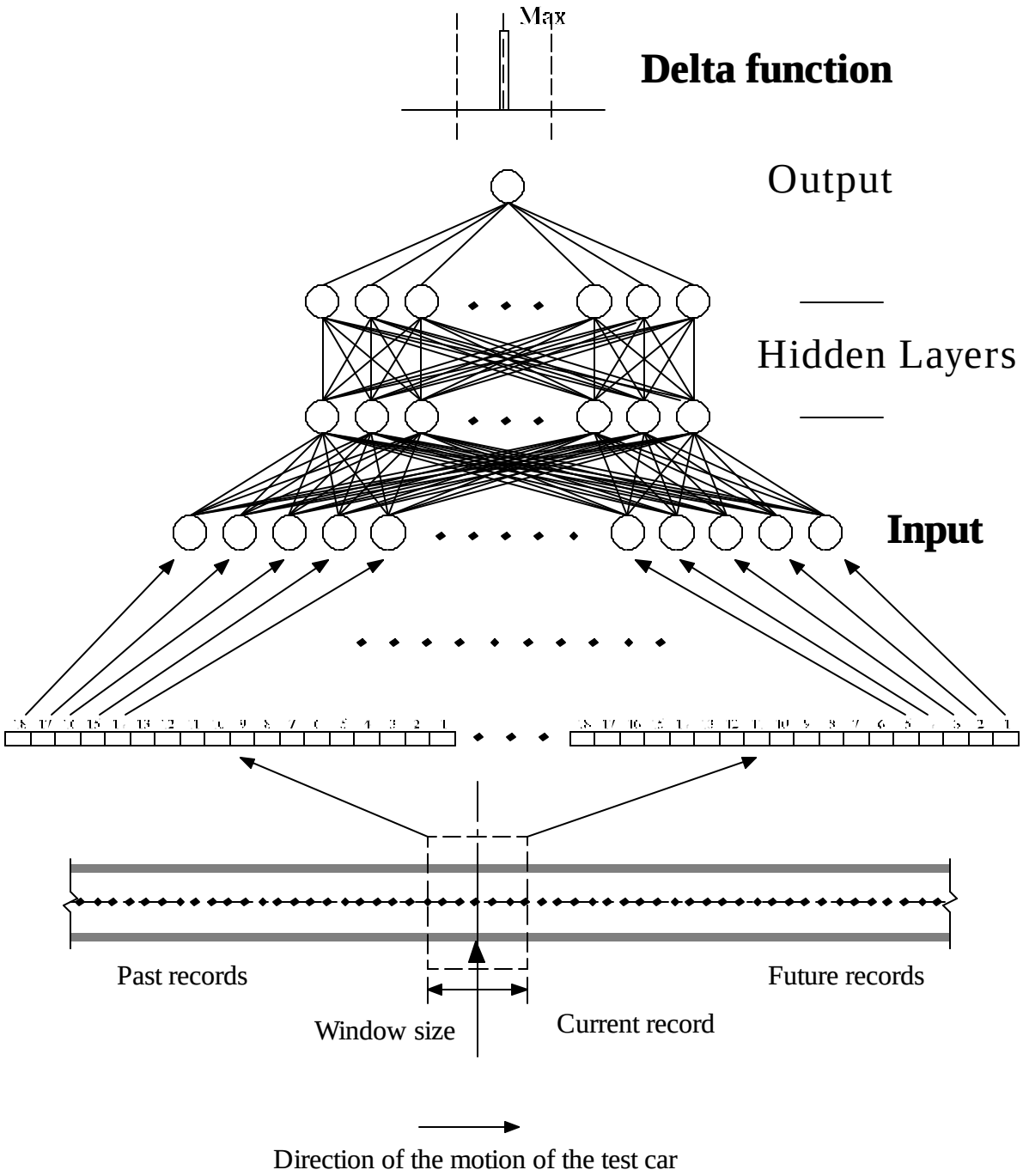


Figure 14. First level defect detection neural network

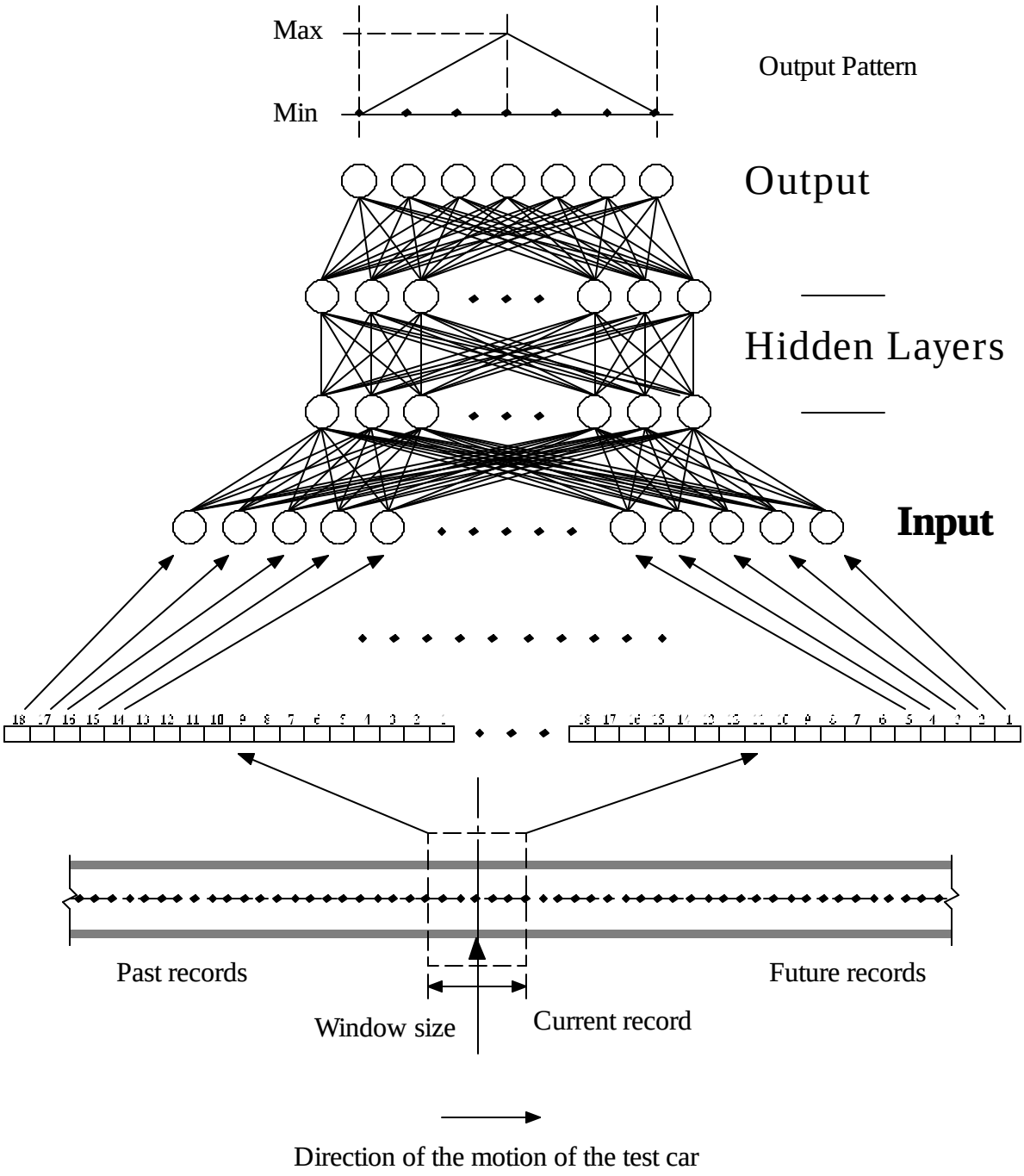


Figure 15. Second level neural network for defect identification

3.2. Neural networks in B-scan method

In the B-Scan data processing system, the 24 channel raw data from UIB are processed into record-by-record data. Each record represents an object created by an algorithm. The record contains information on the channel, gate, object location, object length, start depth, end depth, and the amplitude of the return signal. The object creation process is schematically shown in Figure 16.

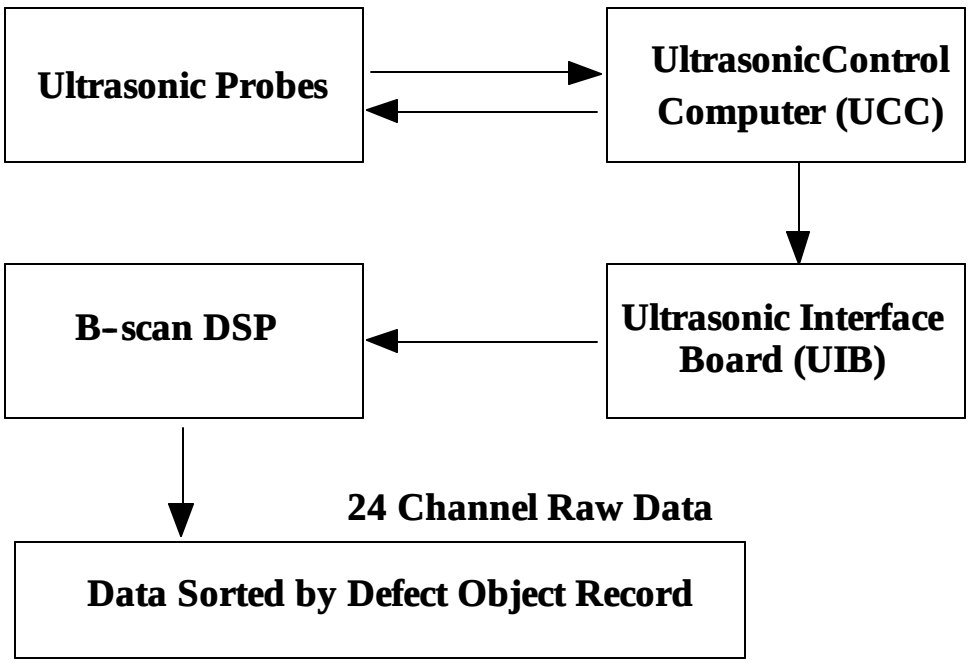


Figure 16. The 24 Channel B-scan raw data

Implementation of Record-by-Record Approach

A B-Scan record is composed of channel and gate numbers, the location (pulse number) of the objects, the object length, start range (depth from the rail top to the ultrasonic reflection surface), end range, and signal strength.

$$B\text{-}\ Scan\ Object\ Record\ = \begin{bmatrix} Channel \text{ and } Gate\ Nos. \\ Location\ of\ the\ Object \\ Object\ Length \\ Start\ Range \\ End\ Range \\ Signal\ Strength \end{bmatrix}$$

Each channel and gate listed in Table 2 creates a record if there is a signal that satisfies the predefined object creation criteria.

The B-Scan neural network input data is prepared for each channel and gate from the B-Scan Object Record Data. The B-Scan neural network input data consist of relative distance between previous objects and current objects, object length, start range, end range, and signal strength. The B-Scan neural network uses the current object records created from signals measured at each channel and gate and the previous histories of object records as well. The rationale for this approach is based on the the fact that one object does not contain enough information for detection and identification of the defect. For example, the same object can be a defect or no defect depending on the neighborhoods of the object.

Table 2. The 24 Channel Data (per Rail) and the Corresponding Probes

Channels		Gate		
1 (13)	zero	0	Web	Head & Web
		3	Depth	Bottom loss
2 (14)	37Fwd	0	Forward 37	Head & Web
		2	CDA	Head & Web
3 (15)	37Rev	0	Reverse 37	Head & Web
		2	CDA	Head & Web
4 (16)	70Fwd	0	CLTD	Head
		2	CDA	Head
5 (17)	70Rev	0	CLTD	Head
		2	CDA	Head
6 (18)	DFGaFwd	0	GageDF	Head (Gage Side)
		2	CDA	Head (Gage Side)
7 (19)	DFGaRev	0	GageDF	Head (Gage Side)
		2	CDA	Head (Gage Side)
8 (20)	DFFdFwd	0	FieldDF	Head (Field Side)
		2	CDA	Head (Field Side)
9 (21)	DFFdRev	0	FieldDF	Head (Field Side)
		2	CDA	Head (Field Side)
10 (22)	0VSH	0	VSH	Head
		2	CDA	Head
11 (23)	VSHGa	0	VSHGage	Head (Gage Side)
		2	CDA	Head (Gage Side)
12 (24)	VSHFd	0	VSHField	Head (Field Side)
		2	CDA	Head (Field Side)

Figure 17 shows the neural network architecture for using the B-Scan Record-by-Record data. Neural network input nodes are connected to 12 channel and gate sets per rail and each set of channel and gate includes the current and n-previous history data of object records.

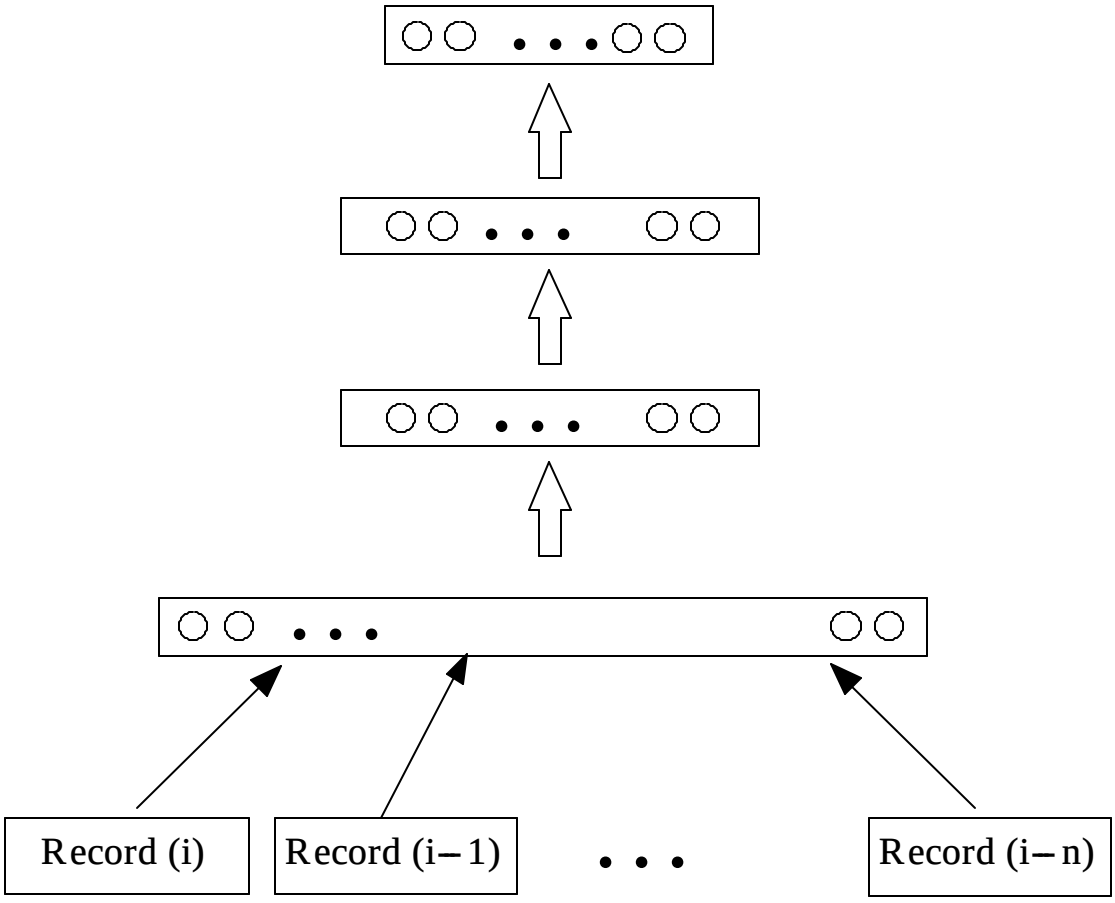


Figure 17. Neural network architecture using B-scan object record

The number of input nodes depends on how many channels and history data are used in the system. In this case, each record is composed of five data as explained previously (the relative distance to the neighboring object, object length, start range, end range, and signal strength), and 3 previous history records are used for 12 channels & gates (shaded area in Table 16). The

total number of neural network input nodes is 260. This number is calculated as follows: the current plus 3 previous records (4) multiplied by 5 data per record multiplied by 13, the number of channels.

Verification with Simulated Defects Data

Initially a simple record-by-record data set is created and used to verify the proposed methodology. The data consists of the relative distance between previous objects and current objects, the object length, start depth and end depth as shown in Figure 18. The current object and two previous object records are used as the input for the neural network. Consequently, 12 input nodes are used in the input layer of neural network. The output layer has two nodes. The neural networks are trained to return ones for the output nodes if a defect is detected among the three object, otherwise the output values are zeros. The neural network used in the verification example is shown in Figure 19.

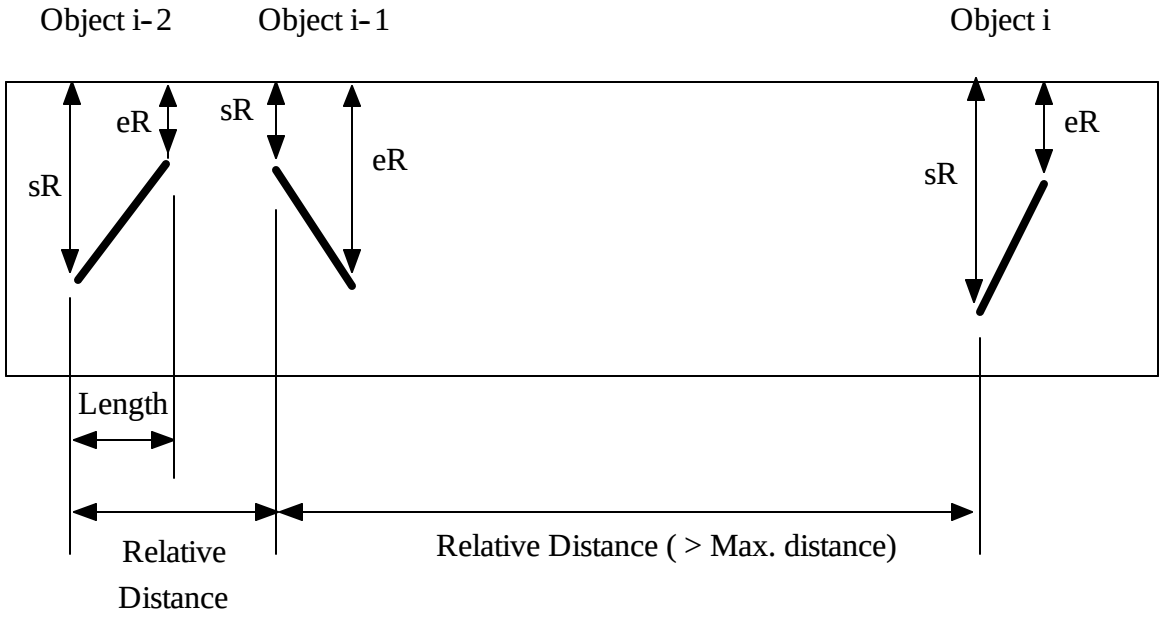


Figure 18. Elements of the simulated B-scan data.

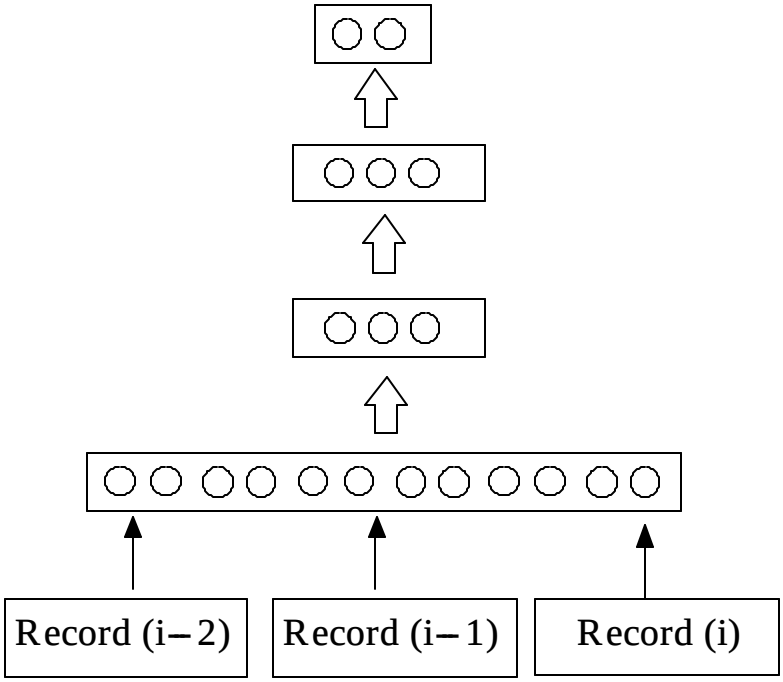


Figure 19. Neural network architecture for the simulated B-scan Data

The B-scan objects are created whenever a reflected signals is received. As a results, the objects can be at long distances from each other. The previous objects that are too far from the current location of the sensors can not, and should not, affect the current defect detection. This observation is implemented by limiting the distance for including the previous objects; if the previous objects are further away than a pre-defined distance from the current object then they are ignored and not used as input to the neural network.

The B-scan neural network was trained with 34 training data sets shown in Table 3 that included a total of 68 objects. The objects shown as shaded in the table were not included in the training data. They were used for testing of the trained neural network. The 33 testing input sets included nine objects; object numbers 36, 39, 42, 45, 54, 57, 62, 65, and 68.

The test results are compared with the expected B-scan NN output in Table 4. Gray cells indicates failure in detecting a defect in the neural network input. However, all the defects were successfully detected, and there were no false positive, as shown in last two columns in the table.

Table 3. Object numbers used to identify pattern (Test Case)

Pattern Number	Objects used for NN			Pattern Number	Objects used for NN		
	i-2	i-1	i		i-2	i-1	i
1	34	35	36	17	50	51	52
2	35	36	37	18	51	52	53
3	36	37	38	19	52	53	54
4	37	38	39	20	53	54	55
5	38	39	40	21	54	55	56
6	39	40	41	22	55	56	57
7	40	41	42	23	56	57	58
8	41	42	43	24	57	58	59
9	42	43	44	25	58	59	60
10	43	44	45	26	59	60	61
11	44	45	46	27	60	61	62
12	45	46	47	28	61	62	63
13	46	47	48	29	62	63	64
14	47	48	49	30	63	64	65
15	48	49	50	31	64	65	66
16	49	50	51	32	65	66	67
				33	66	67	68

Table 4. Comparison between expected NN outputs and results (Test Case)

Pattern Number	Expected NN Output		Actual NN Output		Defect Number	Defect Detected?
	Output 1	Output 2	Output 1	Output 2		
1	1	1	0.988571	0.988629	36	yes
2	1	1	1.007975	1.00803		
3	1	1	0.993009	0.993015		
4	1	1	1.000342	1.000412	39	yes
5	1	1	1.007099	1.007164		
6	1	1	-4.73E-02	-4.73E-02		
7	1	1	0.999304	0.999369	42	yes
8	1	1	1.000596	1.000676		
9	1	1	1.005876	1.005868		
10	1	1	1.0037	1.003759	45	yes
11	1	1	1.011168	1.011065		
12	1	1	-7.73E-02	-7.73E-02		
13	0	0	-0.00077	-0.00077		No Defects
14	0	0	1.24E-04	1.32E-04		
15	0	0	-0.00045	-0.00045		
16	0	0	-0.00022	-0.00021		
17	0	0	1.16E-03	1.17E-03		
18	0	0	1.24E-04	1.32E-04		
19	1	1	-7.34E-02	-7.34E-02	54	yes
20	1	1	1.0104	1.010384		
21	1	1	0.992426	0.992431		
22	1	1	-4.09E-04	-4.06E-04	57	yes
23	1	1	1.009127	1.009151		
24	1	1	0.987229	0.987248		
25	0	0	1.39E-04	1.43E-04		No Defects
26	0	0	-0.00019	-0.00019		
27	1	1	0.953297	0.953407	62	yes
28	1	1	1.012898	1.012708		
29	1	1	7.89E-04	7.38E-04		
30	1	1	1.000946	1.000795	65	yes
31	1	1	0.997244	0.997163		
32	1	1	0.939121	0.939219		
33	1	1	1.012898	1.012708	68	yes

A set of neural networks were developed and applied to actual test track B-scan data. The results indicated a good performance of the trained neural networks.

4. Planned technical approach

The technical approach in this project involved two major steps that could only be performed sequentially. The first step was primarily collection, digitization, and storing of unprocessed ultrasonic data in the laboratory and on the test track in Danbury, Connecticut. The second step was using the collected data to develop, train, test, and evaluate a set of neural networks. This project was jointly funded by TRB and Sperry Rail Service. The data collection was to be done at Sperry Rail Service at Danbury, CT, while development, training and testing of neural networks was to be carried out at University of Illinois at Urbana-Champaign.

The first task in the proposal dealt with developing methods for collecting the unprocessed ultrasonic data. The volume of the unprocessed data is very large and it is not stored in Sperry's detection cars, only the processed data gets stored. The following paragraph is the text of the first task in the proposal.

> "Work with the technical staff at Sperry Rail Service to develop a method of collecting and storing the unprocessed ultrasonic data onboard their detector cars. Conduct a number of test runs with the detector cars on Sperry's test track at Danbury Connecticut and collect both processed data and unprocessed data. The flaws in the test track have been carefully mapped and can be easily located in the data stream."

At the early stages of this project we intended to establish a rail flaw detection laboratory at the University of Illinois at Urbana-Champaign with loaned equipment from Sperry Rail Service. The following paragraph is the text of the second task in the proposal.

> "Establish a rail flaw detection laboratory at the University of Illinois at Urbana-Champaign. This laboratory will be set up mostly with the loaned equipment from Sperry and AAR, including sensor wheel sets, individual sensor sets, electronic display and recording equipment, and sections of rail with known flaws. Initially, a number of tests will be performed to record unprocessed data for individual flaws. Later, in the course of the pro-

posed research, the laboratory will be used for performing verification tests. The laboratory will provide a control environment to evaluate the intermediate steps in the process of developing a neural network flaw detection system."

The purpose of establishing the laboratory was to identify the main characteristics of the unprocessed data for each class of rail flaw and to identify the significant components for use in the neural networks. Initially, it was intended that two laboratories would be established: one at the Department of Civil and Environmental Engineering at the University of Illinois at Urbana-Champaign, and the second laboratory at the Sperry Rail Service. After the compilation of the list of the equipment and pricing them, it was decided that only one laboratory would be established at the Sperry Rail Service. This would have the added advantage of eliminating the need for the costly shipment of the rail specimens with flaws from Danbury to Urbana.

After completing the collection, digitization and storing of the unprocessed ultrasonic data we intended to develop a set of neural networks for rail flaw detection. The following two paragraphs are the text of the last two tasks in the proposal that describe the procedure for developing the neural networks.

"Study the unprocessed data generated in the field and in the laboratory, and develop several candidate systems of neural networks for rail flaw detection. Train the neural networks and evaluate their performance in the laboratory. Select a specific neural network system."

"Further develop the selected neural network system and train the relevant neural networks in the system. Conduct an extensive series of tests and evaluate the performance of the system. Record additional data on the test track and perform blind tests. This task will require several cycles of improving the neural network design, retraining it, and evaluating its performance."

Before any unprocessed data could be collected, Sperry Rail Service withdrew from the project for internal reasons and the project could not be completed.

5. Work accomplished to date

Initially, it was intended that a special laboratory would be set up at Sperry Rail Service for collection, digitization and storing of unprocessed data. A similar, but smaller laboratory was intended to be set up at Urbana at University of Illinois with the loaned equipment and rail sections from Sperry. It was subsequently decided that only one laboratory would be established at Danbury and the Principal Investigator and his graduate students will travel to Danbury as needed.

Sperry Rail Service in consultation with the Principal Investigator completed the selection and purchase of the equipment for the Rail Flaw Detection Laboratory. The following is a partial list of equipment and software that was acquired.

1. UT 340 (standard) pulser/receiver from UTEX Scientific, Canada.

2. Digitizer with 100 MHz sampling and 12 bits.

3. Winspect software that is used to control motion, operate pulser/receiver, acquire data, and process data for different uses. The Winspect has a capability to acquire a full digitized wave-form that is used to extract features for artificial intelligence.

4. I3D software that is used to study wave-propagation within a medium for variety of transducer configurations.

Sperry Rail Service has also loaned some existing equipment to the laboratory. They include platforms for mounting the rail segments for ultrasonic testing, various wheel sets with different transducers, laboratory device for rolling the wheel set over the rail, handset ultrasonic transducers, and a digital oscilloscope.

At that point the plans called for starting the testing in the near future. In the first phase of the testing the Principal Investigator and a graduate student planned to spend about two weeks in Danbury to perform the tests. Thereafter, they planned to travel to Danbury for additional tests as needed.

We also started planning for the second phase of the project, which involved collection of the digitized unprocessed data. Currently, only the processed data is stored. The volume of the unprocessed data is very large and no provisions had been made for storing the unprocessed data in the current electronic equipment on the detection cars. Special equipment needed to be designed and installed on a selected car to digitize and store the unprocessed data. Initially, it was intended that the unprocessed data would be acquired from the runs over the Sperry's test track in Danbury, CT. The location and type of the flaws in the test track are known. Additional data were to be acquired from the revenue runs n the next phase of testing.

After an initial study, it was determined that the collection of unprocessed field data would have required costly design and modification of the electronic systems in the Sperry detection cars. A less costly method was possible for generating the necessary data for this project. A decision was made that for the purpose of this project, the unprocessed data would be collected in the laboratory, duplicating the conditions in the field as closely as possible. Sections of rail with known flaws were to be used and subjected to the same ultrasonic transducer wheel sets that are used in the field. A request for modification of the original proposal was filed with TRB and this modification was subsequently approved.

After a period of uncertainty about the continuation of the project, Sperry decided to continue with the project. During a visit by the Principle Investigator to Sperry Rail Service Laboratory in Danbury, CT a small sample of unprocessed ultrasonic data was collected using the new laboratory set up and the new equipment acquired for this purpose. Shortly after that, Sperry Rail Service decided to withdraw from this project. Without their participation the project could not be completed.

6. Preliminary conclusions and reason for termination of this research project

Even though this project could not be completed, the potential advantage of using unprocessed ultrasonic data remains. Earlier research performed by the Principle Investigator clearly

demonstrated the benefits of using neural networks in ultrasonic rail flaw detection with processed data. It is only logical to conclude that using unprocessed data, that contains far more information than the processed data, will improve the performance of rail flaw detection. This observation is equally shared by the Principle Investigator and Sperry Rail Service.

The necessary neural networks and the procedure for carrying out this development are already in place. In order to proceed to the next step in this project we needed recorded unprocessed data. The recorded unprocessed data should cover the range of all the major types of rail flaws. The data should also include the cases where there are signals but there are no flaws. This data was intended for use in training of neural networks. This would have been followed by testing and evaluation of the trained neural networks. The procedure to be employed in this project was similar to the procedure used in the earlier research with the processed data.

7. Recommendations for future research

The Principle Investigator firmly believes that there is considerable potential in using the unprocessed data to improve the performance of the ultrasonic rail flaw detection. Neural networks can play a central role in realizing this potential. It is highly recommended that this project be continued at a future time when it becomes possible to collect and digitize the necessary unprocessed ultrasonic data.

The future research can be envisioned in two major phases. The first phase will be similar to the research that was planned in this project. In this phase neural network based rail flaw detection methods will be developed for the current configuration of the ultrasonic transducers.

Once effective neural networks have been developed to use the unprocessed data, it will be possible to examine the basics of the ultrasonic rail flaw detection. As part of the this re-examination it would be appropriate to revisit the ultrasonic transducer arrangements currently used in rail flaw detection. The current arrangements of the ultrasonic transducers are designed to generate data for use by the operator and this imposes certain limitations on the design of the whole system. These limitation could be relaxed with the neural network based rail flaw detec-

tion. Neural networks bring additional capabilities to rail flaw detection in terms of speed and the volume of data that they process. Future research can be directed at investigating more effective ways of deploying the ultrasonic transducers in a neural-network-based rail flaw detection.

8. Bibliography

M. R. Banan, J. Ghaboussi and R. L. Florom, "Neural Networks in Railway Engineering: Acoustic Wayside Fault Detection", Proceedings, International Conference on Artificial neural Networks in Engineering, St Louis, November 1994.

J. Ghaboussi, M. R. Banan and R. L. Florom, "Application of Neural Networks in Acoustic Wayside Fault Detection in Railway Engineering", Proceedings, World Congress on Railway Research, Paris France, November 1994.

A. Joghataie, J. Ghaboussi and X. Wu, "Learning and Architecture Determination Through Automatic Node generation", Proceedings, International Conference on Artificial neural Networks in Engineering, St Louis, November 1995.

J. Ghaboussi, M. Zhang, X. Wu and D. A. Pecknold, "Nested Adaptive Neural Networks: A New Architecture", Proceedings, International Conference on Artificial Neural Networks in Engineering, St. Louis, Mo., Nov. 1997.

J. H. Chou, J. Ghaboussi and R. Clark, "Application of Neural Networks to the Inspection of Railroad Rail", Proceedings, Twenty-Fifth Annual Conference on Review of Progress in Quantitative Nondestructive Evaluation, Snowbird Utah, July 1998.

J. Ghaboussi and X. Wu, "Soft Computing with Neural Networks for Engineering Applications: Fundamental Issues and Adaptive Approaches", International Journal of Structural Engineering and Mechanics, vol. 6, No. 8, pp 955 -969, Dec. 1998.

J. Ghaboussi, "Biologically Inspired Soft Computing Methods in Structural Mechanics and Engineering", International Journal of Structural Engineering and Mechanics, v. 11, n. 5, 485-502., April, 2001.

J. H. Chou and J. Ghaboussi, "Genetic Algorithm in Structural Damage Detection", Computers and Structures, v. 79, pp 1335-1353, June 2001.

J. Ghaboussi, "An Overview of Biologically-inspired Soft Computing Methods in Computational Mechanics", Proceedings, World Congress on Computational Mechanics, Vienna, Austria, July, 2002, keynote lecture.